FE

平成29年度　春期
基本情報技術者試験
午後　問題

試験時間	13:00 ～ 15:30（2時間30分）

注意事項

1. 試験開始及び終了は，監督員の時計が基準です。監督員の指示に従ってください。
2. 試験開始の合図があるまで，問題冊子を開いて中を見てはいけません。
3. **答案用紙への受験番号などの記入は，試験開始の合図があってから始めてください。**
4. 問題は，次の表に従って解答してください。

問題番号	問1	問2～問7	問8	問9 ～ 問13
選択方法	必須	4問選択	必須	1問選択

5. 答案用紙の記入に当たっては，次の指示に従ってください。
   (1) 答案用紙は光学式読取り装置で読み取った上で採点しますので，B又はHBの黒鉛筆で答案用紙の**マークの記入方法**のとおりマークしてください。マークの濃度がうすいなど，**マークの記入方法**のとおり正しくマークされていない場合は読み取れません。特にシャープペンシルを使用する際には，マークの濃度に十分注意してください。訂正の場合は，あとが残らないように消しゴムできれいに消し，消しくずを残さないでください。
   (2) **受験番号欄**に**受験番号**を，**生年月日欄**に**受験票の生年月日**を記入及びマークしてください。答案用紙の**マークの記入方法**のとおり記入及びマークされていない場合は，採点されないことがあります。生年月日欄については，受験票の生年月日を訂正した場合でも，訂正前の生年月日を記入及びマークしてください。
   (3) **選択した問題**については，次の例に従って，**選択欄**の**問題番号**の（選）をマークしてください。答案用紙の**マークの記入方法**のとおりマークされていない場合は，採点されません。問2～問7について5問以上マークした場合は，はじめの4問を採点します。問9～問13について2問以上マークした場合は，はじめの1問を採点します。

〔問3，問4，問6，問7，問9を選択した場合の例〕

   (4) **解答**は，次の例題にならって，**解答欄**にマークしてください。答案用紙の**マークの記入方法**のとおりマークされていない場合は，採点されません。

〔例題〕　次の［　　］に入れる正しい答えを，解答群の中から選べ。

春の情報処理技術者試験は，［ a ］月に実施される。

解答群　ア 2　　イ 3　　ウ 4　　エ 5

正しい答えは“ウ 4”ですから，次のようにマークしてください。

例題	a	㋐ ㋑ ● ㋓ ㋔ ㋕ ㋖ ㋗ ㋘ ㋙

裏表紙の注意事項も，必ず読んでください。

〔問題一覧〕

●問 1（必須問題）

問題番号	出題分野	テーマ
問 1	情報セキュリティ	ファイルの安全な受渡し

●問 2～問 7（6 問中 4 問選択）

問題番号	出題分野	テーマ
問 2	ハードウェア	温度モニタ
問 3	データベース	住民からの問合せに回答するためのデータベース
問 4	ネットワーク	無線 LAN におけるデータの送信
問 5	ソフトウェア設計	購買システムにおける注文書出力処理
問 6	プロジェクトマネジメント	プロジェクトの要員計画
問 7	システム戦略	在庫補充方法の変更

●問 8（必須問題）

問題番号	出題分野	テーマ
問 8	データ構造及びアルゴリズム	最短経路の探索

●問 9～問 13（5 問中 1 問選択）

問題番号	出題分野	テーマ
問 9	ソフトウェア開発（C）	マーク式試験の答案の採点
問 10	ソフトウェア開発（COBOL）	売上集計
問 11	ソフトウェア開発（Java）	電気料金プランの比較
問 12	ソフトウェア開発（アセンブラ）	多倍長演算（64 ビット加算，32 ビット乗算）
問 13	ソフトウェア開発（表計算）	宅配便の料金計算

共通に使用される擬似言語の記述形式

擬似言語を使用した問題では，各問題文中に注記がない限り，次の記述形式が適用されているものとする。

〔宣言，注釈及び処理〕

記述形式		説明
○		手続，変数などの名前，型などを宣言する。
/* 文 */		文に注釈を記述する。
処理	・変数 ← 式	変数に式の値を代入する。
	・手続(引数，…)	手続を呼び出し，引数を受け渡す。
	▲ 条件式 ↕ 処理 ▼	単岐選択処理を示す。 条件式が真のときは処理を実行する。
	▲ 条件式 ↕ 処理 1 ┼ ↕ 処理 2 ▼	双岐選択処理を示す。 条件式が真のときは処理 1 を実行し，偽のときは処理 2 を実行する。
	■ 条件式 ↕ 処理 ■	前判定繰返し処理を示す。 条件式が真の間，処理を繰り返し実行する。
	■ ↕ 処理 ■ 条件式	後判定繰返し処理を示す。 処理を実行し，条件式が真の間，処理を繰り返し実行する。
	■ 変数：初期値，条件式，増分 ↕ 処理 ■	繰返し処理を示す。 開始時点で変数に初期値（式で与えられる）が格納され，条件式が真の間，処理を繰り返す。また，繰り返すごとに，変数に増分（式で与えられる）を加える。

〔演算子と優先順位〕

演算の種類	演算子	優先順位
単項演算	＋，－，not	高
乗除演算	×，÷，％	↑
加減演算	＋，－	
関係演算	＞，＜，≧，≦，＝，≠	
論理積	and	↓
論理和	or	低

注記　整数同士の除算では，整数の商を結果として返す。％演算子は，剰余算を表す。

〔論理型の定数〕

true，false

次の問1は必須問題です。必ず解答してください。

問1　ファイルの安全な受渡しに関する次の記述を読んで，設問1～3に答えよ。

情報システム会社のX社では，プロジェクトを遂行する際，協力会社との間で機密情報を含むファイルの受渡しを手渡しで行っていた。X社は，効率化のために，次期プロジェクトからは，インターネットを経由してファイルを受け渡すことにした。

X社で働くAさんは，ファイルを受け渡す方式について検討するように，情報セキュリティリーダであるEさんから指示された。Aさんは，ファイルを圧縮し，圧縮したファイルを共通鍵暗号方式で暗号化した上で電子メール（以下，メールという）に添付して送信し，別のメールで復号用の鍵を送付する方式をEさんに提案した。しかし，Eさんから“①Aさんの方式は安全とはいえない”との指摘を受けた。

Aさんは，暗号化について再検討し，圧縮したファイルを公開鍵暗号方式で暗号化してメールに添付する方式をEさんに提案したところ，“その方式で問題はないが，相手の［　a　］を入手する際には，それが相手のものであると確認できる方法で入手する必要がある点に注意するように”と言われた。

設問1　本文中の下線①でEさんから指摘を受けた理由として，最も適切な答えを，解答群の中から選べ。

解答群

ア　圧縮してから暗号化する方式は，暗号化してから圧縮する方式よりも解読が容易である。

イ　圧縮ファイルを暗号化してもファイル名は暗号化されない。

ウ　共通鍵暗号方式は，他の暗号方式よりも解読が容易である。

エ　ファイルを添付したメールと，鍵を送付するメールの両方が盗聴される可能性がある。

設問２　本文中の［　　］に入れる適切な答えを，解答群の中から選べ。

aに関する解答群

ア　共通鍵　　イ　公開鍵　　ウ　ディジタル署名

エ　パスワード　　オ　秘密鍵

設問３　次の記述中の［　　］に入れる正しい答えを，解答群の中から選べ。

次期プロジェクトでは，協力会社であるＰ社，Ｑ社，Ｒ社及びＳ社と協業する。プロジェクトの期間は 12 か月である。A さんは，各協力会社との間でファイルを受け渡す方式について，E さんから次のように指示されたので，更に検討を進めることにした。

〔ファイルを受け渡す方式に関するＥさんからの指示〕

(1)　メールを使用する方式以外も検討すること。

(2)　ファイルを受け渡す方式は，協力会社ごとに異なっていてもよい。

(3)　協力会社間ではファイルを受け渡さない。

(4)　ある協力会社との間で，ファイルを受け渡すためにアカウントを登録する必要があるシステムを使う場合，その会社からプロジェクトに参加する社員全員のアカウントを登録すること。

(5)　受け渡すファイルの機密度に合った方式を選択すること。機密度には“低”と“高”の２種類がある。X 社のセキュリティポリシでは，機密度が“高”のファイルを，オンラインストレージサービスを利用して受け渡すことを禁止している。

(6)　費用（初期費用とプロジェクト期間中の運用費用の合計）が最も安い方式を選択すること。

各協力会社の参加人数及び受け渡すファイルの機密度は，表１のとおりである。

表１　協力会社の参加人数及び受け渡すファイルの機密度

協力会社	参加人数（人）	受け渡すファイルの機密度
P社	10	“低”だけ
Q社	5	“低”と“高”
R社	50	“低”だけ
S社	25	“低”と“高”

Aさんは，ファイルを受け渡す方式として，次の三つの候補を検討した。

〔ファイルを受け渡す方式の候補〕

(1)　VPNとファイルサーバ

X社の拠点と協力会社の拠点との間でVPN環境を構築し，ファイルを受け渡すためのファイルサーバをX社に設置する。協力会社ごとに，異なるVPN環境の構築と異なるファイルサーバの設置を行う。この方式では，一つの協力会社につき，初期費用としてVPN環境の構築とファイルサーバの設置に100,000円，運用費用としてファイルサーバの運用及びVPN利用に，合わせて月額50,000円が掛かる。初期費用，運用費用ともに利用者数の多寡による影響はない。

(2)　オンラインストレージサービス

インターネット上で提供されているオンラインストレージサービスを利用してファイルを受け渡す。このサービスは，利用者にHTTP over TLSでのアクセスを提供しており，ファイルを安全に受け渡せる。この方式では，初期費用は掛からないが，運用費用として利用者１人当たり月額500円が掛かる。X社では，全社員がこのサービスを利用することにしたので，X社の社員についての運用費用はこのプロジェクトの費用には含めない。

(3)　暗号化機能付きメールソフト

公開鍵暗号方式を使った暗号化機能付きメールソフトを導入し，メールにファイルを添付して受け渡す。この方式を安全に運用するためには，導入時にプロジェクトの参加者全員に対して，メールソフトの利用方法などに関する研修が必要である。この方式では，初期費用として，メールソフトの導入及び研修

に，利用者 1 人当たり 30,000 円が掛かるが，運用費用は掛からない。X 社では，全社員がこのメールソフトを利用することにしたので，X 社の社員についての初期費用はこのプロジェクトの費用には含めない。

A さんは，各協力会社との間でファイルを受け渡す方式について，E さんからの指示に基づき協力会社ごとに選択すべき方式を検討した。その結果と，費用（初期費用とプロジェクト期間中の運用費用の合計）を，表 2 に示す。

表 2　選択すべき方式とその費用

協力会社	選択すべき方式	費用（円）
P 社	オンラインストレージサービス	60,000
Q 社	b	
R 社		
S 社	c	d

注記　網掛けの部分は表示していない。

b，c に関する解答群

ア　VPN とファイルサーバ

イ　オンラインストレージサービス

ウ　暗号化機能付きメールソフト

d に関する解答群

ア　30,000　　イ　60,000　　ウ　150,000

エ　300,000　　オ　700,000　　カ　750,000

キ　1,500,000

次の問２から問７までの６問については，この中から４問を選択し，選択した問題については，答案用紙の選択欄の(選)をマークして解答してください。

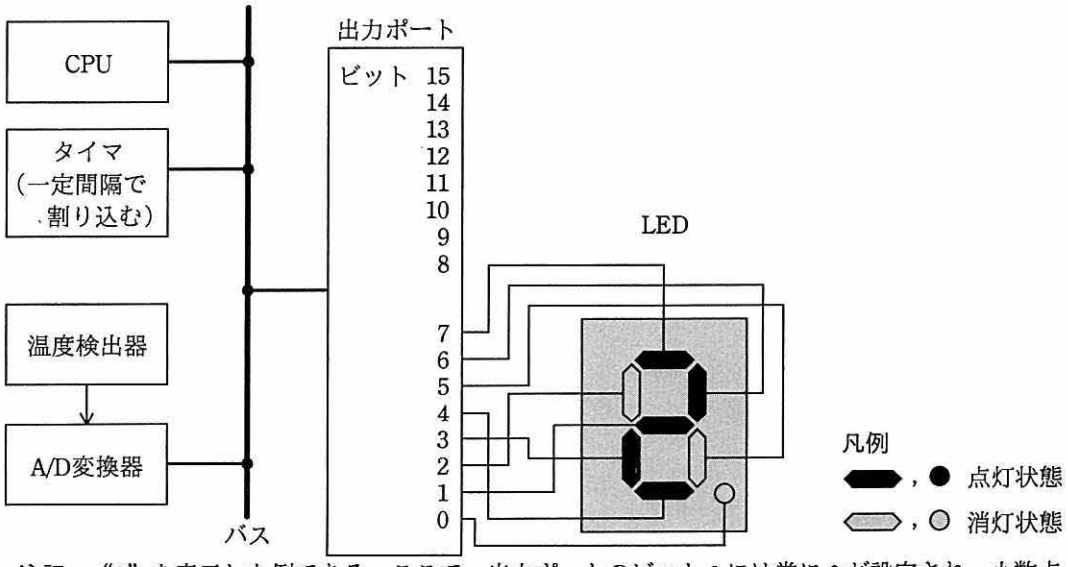
なお，５問以上マークした場合には，はじめの４問について採点します。

問２　温度モニタに関する次の記述を読んで，設問１，２に答えよ。

図１に，ある温度モニタのシステム構成図の一部を示す。

タイマ割込み発生時に起動される割込みプログラムが，温度検出器の出力値を A/D 変換器を介して取り込み，対応した値を７セグメント LED（以下，LED という）に表示するシステムである。温度の検出範囲は 0～70℃とし，検出された温度（以下，検出温度という）を８段階のレベルに対応させて，“1”～“8”の数字で表示する。割込みプログラムが起動されてから表示処理の完了までの時間は，タイマ割込み間隔に比べて十分に短いものとする。

LED の各セグメントは，対応する出力ポートのビットの値が１のとき点灯し，０のとき消灯する。LED に“1”～“8”の数字を表示するために，数字の字形に合わせて，対応する８ビットのデータを出力ポートのビット０～７に設定する。このデータを形状データといい，出力ポートのビット７を最上位ビットとする２桁の 16 進数で表記する。



注記　“2”を表示した例である。ここで，出力ポートのビット０には常に０が設定され，小数点を表示するセグメントは消灯している。

図１　温度モニタのシステム構成図の一部

設問1　次の記述中の［　　］に入れる正しい答えを，解答群の中から選べ。

検出温度は，A/D 変換器によって，表 1 に示す 10℃ ごとの温度基準値の中の最も近い値に近似される。温度基準値の刻み幅の中央の値は高い側の温度基準値に近似される。例えば，検出温度 14℃ は 10℃ に，15℃ は 20℃ に，16℃ は 20℃ に近似される。温度基準値，A/D 変換器出力，レベルの関係を，表 1 に示す。

表 1　温度基準値，A/D 変換器出力，レベルの関係

温度基準値（℃）	A/D 変換器出力（2 進表記）	レベル
0	000	1
10	001	2
⋮	⋮	⋮
70	111	8

LED の表示が図 2 のとき，出力ポートに設定されている形状データは［ a ］，A/D 変換器出力（2 進表記）は［ b ］，検出温度の範囲は［ c ］である。

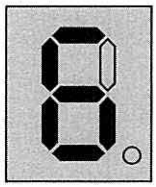


図 2　LED の表示

a に関する解答群

ア　66　　イ　BE　　ウ　DA　　エ　F2　　オ　FE

b に関する解答群

ア　010　　イ　011　　ウ　100　　エ　101　　オ　110

cに関する解答群

ア　40℃ 以上 50℃ 未満　　イ　45℃ 以上 55℃ 未満　　ウ　50℃ 以上 60℃ 未満

エ　55℃ 以上 65℃ 未満　　オ　60℃ 以上 70℃ 未満

設問2　タイマ割込み発生時に起動される割込みプログラムについて，その処理の流れを，図 3 に示す。図 3 中の［　　］に入れる正しい答えを，解答群の中から選べ。

各機器には I/O ポート番号が割り当てられており，割込みプログラムは，各機器に対する制御，データの読込み及び書込みを，I/O ポート番号を指定して行う。LED に接続されている出力ポートには I/O ポート番号 1 が，A/D 変換器には I/O ポート番号 2 と 3 が，それぞれ割り当てられている。表 2 に LED と A/D 変換器の動作概要を示す。また，図 3 で使用している擬似命令の形式と動作は，表 3 のとおりとする。

表 2　LED と A/D 変換器の動作概要

機器	I/O ポート番号	動作概要
LED	1	番号 1 の I/O ポートに形状データを書き込むと，出力ポートの各ビットに値が設定され，LED の各セグメントの点灯と消灯が行われる。
A/D 変換器	2	番号 2 の I/O ポートに値 1 を書き込むと，A/D 変換が開始される。
		番号 2 の I/O ポートから読み込んだ値が 0 ならば，変換中を示す。
		番号 2 の I/O ポートから読み込んだ値が 0 以外ならば，A/D 変換が完了して出力値が確定していることを示す。
	3	A/D 変換完了後に番号 3 の I/O ポートから読み込むと，A/D 変換器の出力値（0～7）が得られる。

表 3　擬似命令の形式と動作

命令の形式	命令の動作
INPUT　I/O ポート番号	番号で指定した I/O ポートに接続されている機器からデータを読み込み，GR に設定する。
OUTPUT　I/O ポート番号	GR に設定したデータを，番号で指定した I/O ポートに接続されている機器に書き込む。

注記　GR は CPU の 8 ビットのレジスタ

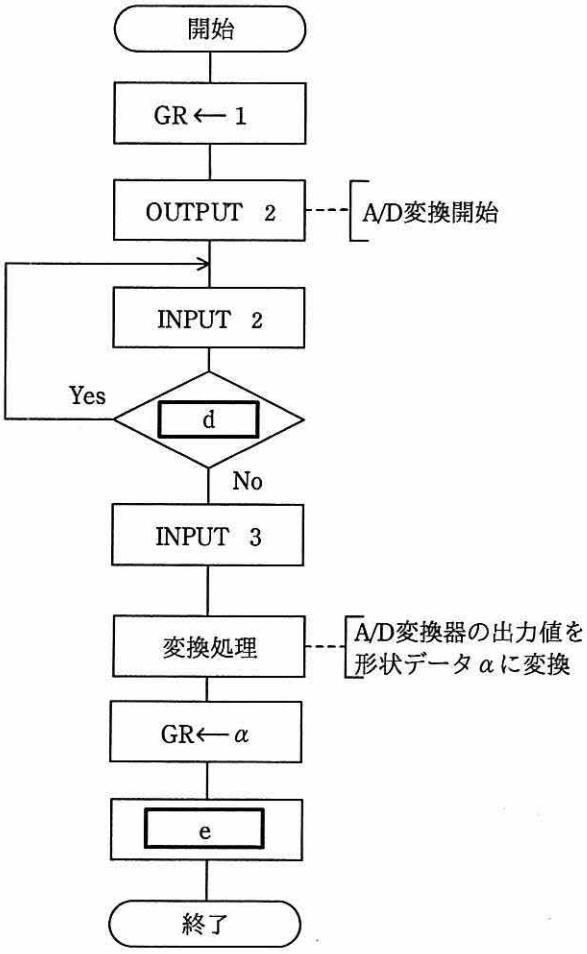


図３　割込みプログラムの処理の流れ

dに関する解答群

ア　GR＜0　　イ　GR＝0　　ウ　GR＝1

エ　GR＞1

eに関する解答群

ア　INPUT 1　　イ　INPUT 2　　ウ　INPUT 3

エ　OUTPUT 1　　オ　OUTPUT 2

選択した問題は，選択欄の(選)をマークしてください。マークがない場合は，採点されません。

問３　住民からの問合せに回答するためのデータベースに関する次の記述を読んで，設問１～３に答えよ。

W 市役所の生活環境課では，職員は住民からのごみの出し方に関する問合せがあったとき，対象物（ごみ）を検索条件としてルール表を検索し，出し方のルールを回答する業務を行っている。

問合せごとに一意な受付 No を割り当て，受付日，回答日，回答したルールの区分 ID 及び対象物 ID を入力したレコードを問合せ記録表に登録する。問合せがあった対象物が対象物表になかった場合，回答日，区分 ID 及び対象物 ID は NULL にして，メモ欄に当該対象物をテキストで入力したレコードを登録する。後日，問合せ記録表から，回答日，区分 ID 及び対象物 ID が NULL のレコードを選択し，メモ欄に入力されている当該対象物の分別区分を決定し，一意な対象物 ID を割り当て，対象物表に新たな対象物のレコードを登録する。ルール表に新たなルールのレコードを登録して，問合せ記録表の選択したレコードに，追加したルールの区分 ID と対象物 ID を設定する。

データベースの表構成とデータ格納例は，図 1 のとおりである。下線付きの項目は主キーを表す。

区分表

区分 ID	分別区分
L0008	可燃物

対象物表

対象物 ID	対象物
S0123	植木の枝

ルール表

区分 ID	対象物 ID	更新日	出し方のルール
L0008	S0123	2015-03-01	乾燥させ，ひもで束にする。

問合せ記録表

受付 No	受付日	回答日	区分 ID	対象物 ID	メモ欄
C003456	2016-05-14	NULL	NULL	NULL	自転車のタイヤチューブ

図１　データベースの表構成とデータ格納例

業務を行っている過程で幾つかの課題が明らかになった。これらのうちから対策すべき課題を選び，それを実現するために，データベースの表構成の見直し案を作成した。その結果を，表1に示す。

表1　対策する課題とデータベースの表構成の見直し案

課題	表構成の見直し案
同じ対象物でも，大きさなどによって出し方を分ける必要が出てきた。	ルール表に主キーとしてルール ID の項目を追加し，区分 ID と対象物 ID の項目を主キーではなくする。問合せ記録表にルール ID の項目を追加する。
新たに登録したルールについて，不都合があり，修正などをしたことが何度かあった。	ルール表に登録状態の項目を追加する。登録状態の項目の値は，“未公開”，“公開”のいずれかである。 理由は，ルール表に新たなルールを追加する場合，住民からの問合せに対する検索の対象とならない（未公開）状態で一旦登録し，課内でレビューした後，問題がなければ検索の対象（公開）とするためである。

設問1　ルール ID の項目を追加する理由として適切な答えを，解答群の中から選べ。

解答群

ア　同じ区分 ID の同じ対象物 ID に対する出し方のルールを複数件登録できるようにする。

イ　同じ区分 ID の異なる対象物 ID に対する出し方のルールを登録できるようにする。

ウ　異なる区分 ID の同じ対象物 ID に対する出し方のルールを登録できるようにする。

設問2　次の記述中の［　　］に入れる適切な答えを，解答群の中から選べ。

ルール表のレコードを特定するために，問合せ記録表にルール ID の項目を追加するとき，ルール ID を外部キーとしたルール表に対する［　a　］制約を設定する。追加するルール ID の項目には，［　b　］。

aに関する解答群

ア　非NULL　　イ　NULL　　ウ　UNIQUE

エ　参照　　オ　検査　　カ　主キー

bに関する解答群

ア　非NULL制約は適用できない

イ　UNIQUE制約を適用する

ウ　更新操作を行うことはできない

エ　主キー制約を適用する

設問3　ルール表作成用 DDL の見直しについて，次の記述中の［　　］に入れる適切な答えを，解答群の中から選べ。

図2に示す，ルールIDの項目と登録状態の項目追加後のルール表作成用DDLのレビュー時に，［ c ］は削除すべきとの指摘を受けた。また，登録状態の項目に対する制約が不足しているとの指摘もあり，［ d ］を追加することになった。

```
CREATE TABLE ルール (
  ルールID CHAR(6) PRIMARY KEY,
  区分ID CHAR(5) NOT NULL,
  対象物ID CHAR(5) NOT NULL,
  登録状態 VARCHAR(10) NOT NULL,
  更新日 DATE NOT NULL,
  出し方のルール VARCHAR(2048),
  FOREIGN KEY (区分ID) REFERENCES 区分(区分ID),
  FOREIGN KEY (対象物ID) REFERENCES 対象物 (対象物ID),
  UNIQUE (区分ID, 対象物ID)
)
```

図2　ルール表作成用DDL

cに関する解答群

ア　区分IDの項目及び対象物IDの項目のUNIQUE制約

イ　対象物IDの項目の非NULL制約

ウ　対象物IDの項目の参照制約

エ　ルールIDの項目の主キー制約

dに関する解答群

ア　CHECK(登録状態 IN('未公開','公開'))

イ　CHECK(登録状態 IS NULL)

ウ　UNIQUE('未公開','公開')

エ　WHERE 登録状態 IN('未公開','公開')

選択した問題は，選択欄の(選)をマークしてください。マークがない場合は，採点されません。

問4　無線 LAN におけるデータの送信に関する次の記述を読んで，設問 1～3 に答えよ。

無線 LAN ではフレームを使用してデータを送信する。フレームにはデータフレームや ACK フレームがあり，データはデータフレームに格納して送信する。データフレームには，フレームの制御情報と TCP/IP パケットが含まれる。一つのデータフレームに格納できるデータ長には上限があり，上限を超えるデータを送信したいときは，データを複数のデータフレームに分けて送信する。無線 LAN では，複数の機器から送信されるフレームがなるべく衝突しないように工夫して，フレームを送信する。

無線 LAN におけるデータフレーム送信の大まかな流れを，(1)～(7)に示す（図 1 参照）。ここで，W1，W2 は待機時間を表す量である。W2 はあらかじめ決められた値で，W1 は W2 を超えるように毎回決められるランダムな値である。

(1)　データフレームを送信しようとする機器は，送信前に他の機器がフレームを送信しているかどうかを調べる。

(2)　他の機器がフレームを送信していれば，そのフレームの送信が終了するまで待機する。

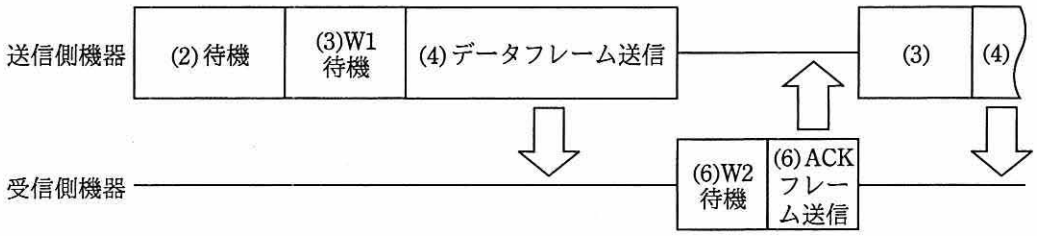
(3)　他の機器がフレームを送信していなければ，W1 だけ待機する。

(4)　W1 の間に，他の機器がフレームの送信を開始しなければ，データフレームを送信し，受信側機器からの ACK フレームの到着を待つ。

(5)　W1 の間に，他の機器がフレームの送信を開始したときは，(2)に戻る。

(6)　データフレームを受信した機器は，データフレームを受信後，W2 だけ待機した後，ACK フレームを送信する。

(7)　複数のデータフレームを送信する必要があるときは，(1)～(6)を繰り返す。



注記　矢印はフレームの送信方向を表す。

図1　データフレーム送信の例

一つのデータフレームに格納できる最大データ長（フレームの制御情報や TCP/IP パケットのヘッダなどの情報を除くデータ本体）を 1,460 バイトとする。物理層の通信速度が 54 M ビット／秒のとき，最大データ長のデータを格納したデータフレームの送信時間は 248 マイクロ秒，ACK フレームの送信時間は 24 マイクロ秒である。W1 の平均と W2 が，物理層の通信速度に関係なく，それぞれ 101.5 マイクロ秒と 16 マイクロ秒であるとき，データ送信速度（単位時間に送信できるデータ量）は，最大で［　a　］M ビット／秒程度になる。

次に，三台の機器 A，B 及び C がある場合について考える。<u>① 機器 A だけが機器 C に対して 1 M バイトのデータの送信処理を行ったとき</u>と，<u>② 機器 A と機器 B のそれぞれが機器 C に対して，それぞれ 1 M バイトのデータの送信処理を同時に開始したとき</u>とを比較すると，機器 A から機器 C へのデータの送信処理に掛かる時間は，平均的には［　b　］。ここで，機器 A と機器 C の間，及び機器 B と機器 C の間それぞれの物理層の通信速度は 54 M ビット／秒とする。また，フレームの衝突はなく，フレームの再送信も行われなかったものとする。

設問 1　本文中の［　　］に入れる正しい答えを，解答群の中から選べ。

a に関する解答群

ア　4　　イ　6　　ウ　30　　エ　47

b に関する解答群

ア　①の方が短い　　イ　②の方が短い　　ウ　等しい

設問2　次の記述中の［　　］に入れる適切な答えを，解答群の中から選べ。ここで，c1 と c2 に入れる答えは，c に関する解答群の中から組合せとして適切なものを選ぶものとする。

無線 LAN の物理層の通信速度は，機器間の距離や障害物の有無などによって変化する。機器 B を機器 C から離すことによって，機器 B と機器 C の間の物理層の通信速度が 24 M ビット／秒になった。機器 A と機器 C の間の物理層の通信速度は 54 M ビット／秒のままである。機器 A と機器 B のそれぞれが，機器 C に対して，それぞれ 1 M バイトのデータの送信処理を同時に開始した。このとき，機器 B と機器 C の間の物理層の通信速度が 54 M ビット／秒から 24 M ビット／秒になったことによって，［ c1 ］。したがって，機器 A から機器 C へのデータ送信速度は，機器 B と機器 C の間の物理層の通信速度が 54 M ビット／秒だったときと比較して，［ c2 ］。

c に関する解答群

	c1	c2
ア	機器 A が，機器 B によるフレーム送信終了まで待機する時間が長くなる	低下する
イ	機器 A が，W1 だけ待機する回数が多くなる	低下する
ウ	機器 A と機器 C の間の物理層の通信速度が，機器 B と機器 C の間の物理層の通信速度と比べて向上する	向上する
エ	機器 A と機器 C の間の物理層の通信速度に与える影響はない	変わらない
オ	機器 B の W1 が長くなり，機器 A の通信が優先される	向上する
カ	送信されるデータの量は変わらない	変わらない

設問３　無線 LAN における，データ送信速度を向上させる工夫を三つ，次に示す。これらのうち，伝送効率（データ送信速度 ÷ 物理層の通信速度）の向上に寄与するものだけを全て挙げたものとして適切な答えを，解答群の中から選べ。

〔データ送信速度を向上させる工夫〕

(Ⅰ)　複数のデータフレームに対応する一つの ACK を返すことによって，ACK フレームの個数を削減する。

(Ⅱ)　複数のデータフレームをまとめて送信することによって，W1 だけ待機する回数を削減する。

(Ⅲ)　物理層の通信速度を向上させる。

解答群

ア　(Ⅰ)	イ　(Ⅰ)と(Ⅱ)	ウ　(Ⅰ)と(Ⅲ)
エ　(Ⅱ)	オ　(Ⅱ)と(Ⅲ)	カ　(Ⅲ)

選択した問題は，選択欄の(選)をマークしてください。マークがない場合は，採点されません。

問5　購買システムにおける注文書出力処理に関する次の記述を読んで，設問1～3に答えよ。

電気製品を製造しているJ社では，製造部門が要求する電子部品（以下，商品という）を購買部門が発注する際に利用する購買システムの構築が進行中である。購買システムを利用するには，Web ブラウザを使用する。J社の購買部門の社員（以下，購買担当者という）は社内ネットワーク経由で，仕入先の営業員はインターネット経由で，それぞれ購買システムを利用する。

〔購買システムを利用した発注業務の説明〕

(1)　発注する全ての商品は，商品ごとに一意になる商品番号で管理する。商品ごとに，仕入先は1社に定まっている。

(2)　購買部門は三つのグループ（以下，購買グループという）に分かれており，購買グループごとに取り扱う商品が異なる。

(3)　各購買担当者は，自グループが取り扱う商品について，製造部門からの要求を取りまとめ，仕入先ごとに分類し，発注登録を行う。このとき，仕入先ごとに，一意の注文番号が割り当てられ，商品ごとに001から始まる注文枝番が連番で割り当てられる。

購買担当者の発注登録，仕入先の営業員の注文書出力及び購買システムの関係は，図1のとおりである。

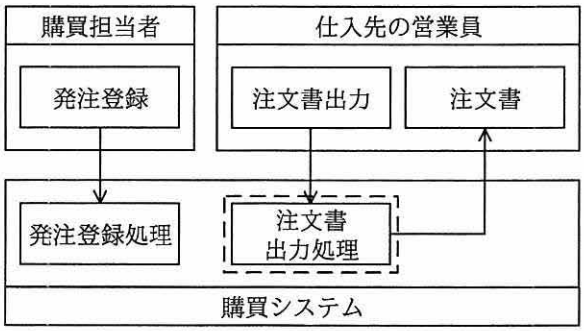


図１　購買担当者の発注登録，仕入先の営業員の注文書出力及び購買システムの関係

購買システムの構築プロジェクトのメンバ K 君は，図 1 の破線で囲まれた注文書出力処理の開発を担当することになった。注文書出力処理は，仕入先の営業員が購買システムから J 社の注文書を出力するための処理である。

〔注文書出力処理の説明〕

(1)　仕入先の営業員が注文書を出力する操作の流れを，図 2 に示す。

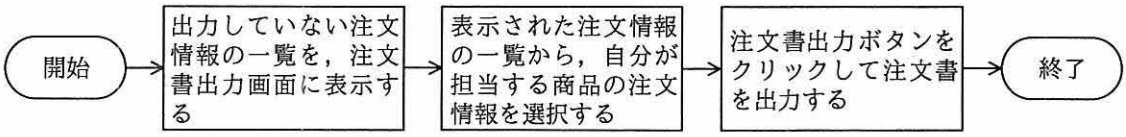


図２　注文書を出力する操作の流れ

(2)　仕入先の端末の注文書出力画面には，まだ注文書として出力していない注文情報だけが，図 3 に示すように注文番号ごとの注文枝番ごとに一覧で表示される。

(3)　購買システムから出力される注文書は，購買グループごとに作成され，注文情報が希望納期，注文番号，注文枝番の昇順に並んでいる。注文書の各ページには，最大 5 件の注文情報が含まれる。注文情報は，購買グループ名，希望納期，注文番号，注文枝番，商品番号，数量，納品場所，購買担当者などを要素としてもつ。

注文書出力画面と注文書の例を，それぞれ図 3 と図 4 に示す。

注文書出力画面

注文書出力

選択	注文番号	注文枝番	商品番号	商品名	数量	希望納期	購買グループ名	購買担当者
□	A20170427001	001	X01SS001	XXXXXX	3	2017-04-27	Aグループ	情報太郎
□	A20170427001	002	Y01ZZ002	YYYYYY	1	2017-04-27	Aグループ	情報太郎
				⋮				
□	C20170512001	001	Z01ZZ001	ZZZZZZ	3	2017-05-12	Cグループ	試験五郎

図３　注文書出力画面の例

注文書　　XXXX-XX-XX

取引先名　XX部品製造株式会社
住所　東京都新宿区 XX-XX

J株式会社
購買グループ名　Aグループ
連絡先　03-9999-9999

注文番号 注文枝番	購買担当者	商品番号	数量	希望納期
		商品名	単価（円）	納品場所
A20170427001 001	情報太郎	X01SS001	3	2017-04-27
		XXXXXX	300	AREA-01
⋮				

最大５件

- 1 -

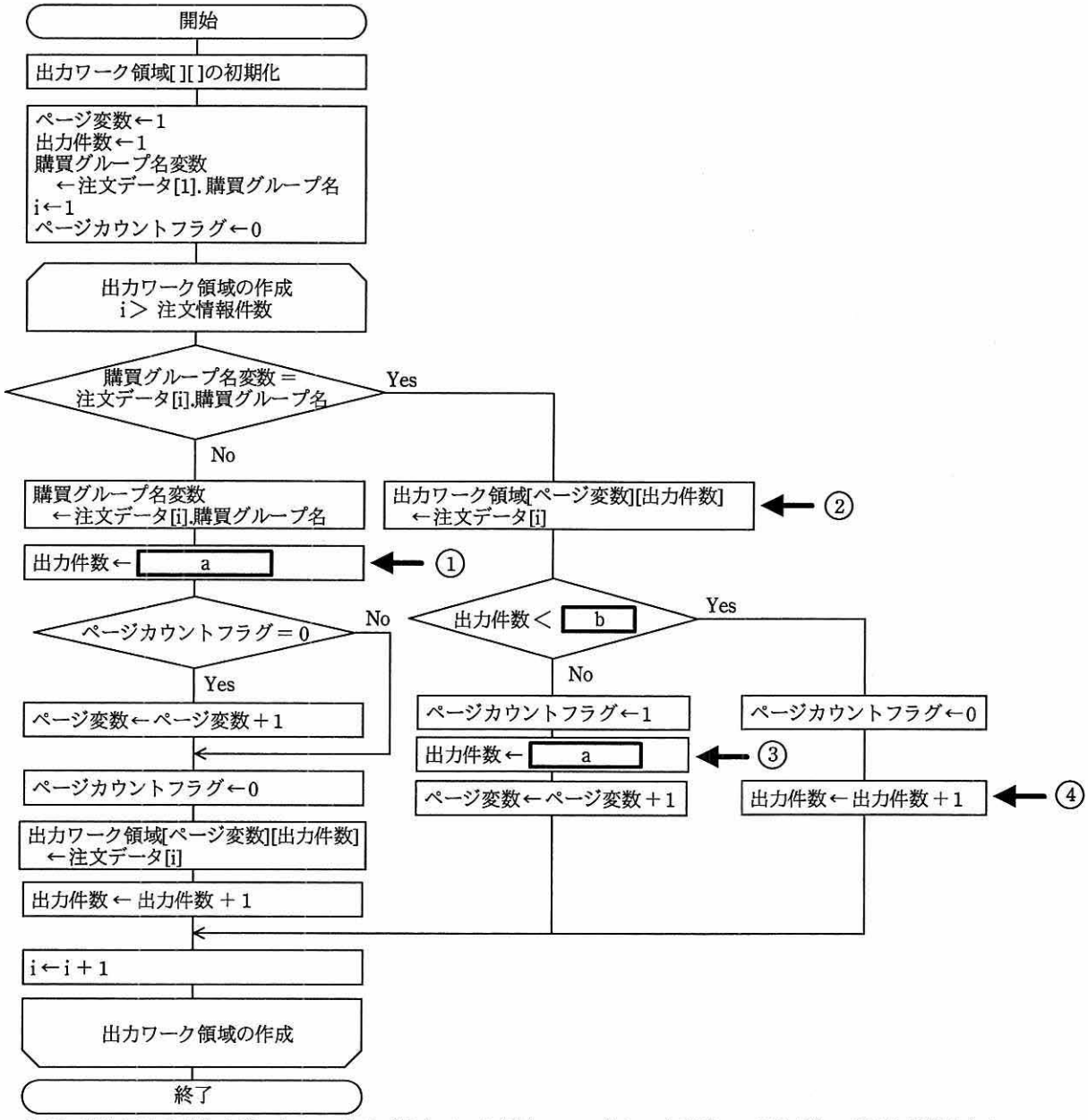
図４　注文書の例

Ｋ君は，注文書出力画面で注文書出力ボタンがクリックされたときに実行する処理の一つである出力準備処理の流れ図を作成した。作成した流れ図を，図５に示す。

出力準備処理とは，注文書に含める注文情報を出力順に並べて出力ワーク領域に格納するものである。出力ワーク領域は，１回の注文書出力処理で出力される注文書に含まれる注文情報を格納する２次元配列である。出力ワーク領域[i][j]は，iが注文書のページを表し，jは注文書のページ内での出力順を表している。iもjも１から始まる。

出力準備処理は処理を開始するとき，注文書出力画面で選択された注文情報を，購買グループ名，希望納期，注文番号，注文枝番の順にそれぞれ昇順に整列して格納した１次元配列である注文データ[]と，注文情報件数を引数で受け取る。ここで，注文

データ[1]には，注文書の1ページ目の第1番目に出力する注文情報が格納されている。



注記　流れ図中の“注文データ[i].xxx”や“出力ワーク領域[i][j].xxx”という記述で，それぞれの配列に格納された注文情報の要素xxxの値を表す。ページカウントフラグは，ページ変数を制御するフラグである。

図5　出力準備処理の流れ図

設問１　図５中の［　　］に入れる正しい答えを，解答群の中から選べ。

a，bに関する解答群

ア　0　　イ　1　　ウ　2　　エ　3

オ　4　　カ　5　　キ　50

設問２　K君は，出力準備処理のホワイトボックステストを実施するために，図６に示す注文データのテストデータを作成した。次の記述中の［　　］に入れる正しい答えを，解答群の中から選べ。

	購買グループ名	希望納期	注文番号	注文枝番	商品番号	数量	購買担当者
注文データ[1]	Aグループ	2017-04-27	A20170427001	001	X01SS001	3	情報太郎
注文データ[2]	Aグループ	2017-04-27	A20170427001	002	Y01ZZ002	1	情報太郎
注文データ[3]	Aグループ	2017-04-30	A20170430001	001	D01SS001	3	情報太郎
注文データ[4]	Bグループ	2017-05-10	B20170510001	001	V01ZZ002	1	技術花子
注文データ[5]	Bグループ	2017-05-11	B20170511001	001	V01ZZ002	1	技術花子
注文データ[6]	Cグループ	2017-05-12	C20170512001	001	Y01ZZ001	3	試験五郎

注記　注文情報の一部の要素は省略してある。

図６　注文データのテストデータ

図６の注文データのテストデータを利用して，出力準備処理をテストしたときに，図５の［ c ］の処理が行われなかった。K君は，全ての処理を実行するために，［ d ］。

cに関する解答群

ア　①　　イ　②　　ウ　③　　エ　④

dに関する解答群

ア　購買グループ名がAグループである注文情報を１件追加した

イ　購買グループ名がBグループである注文情報を３件追加した

ウ　購買グループ名がCグループである注文情報を２件追加した

エ　購買グループ名がCグループである注文情報を３件追加した

設問３　出力準備処理のブラックボックステストを行う場合，Ｋ君が実施すべきこととして適切な答えを，解答群の中から選べ。

解答群

ア　流れ図の全ての分岐処理が実行されたことを確認する。

イ　入力として渡した引数の注文データ，注文情報件数の内容が仕様どおりであることを確認する。

ウ　入力として渡した引数の内容に対して出力ワーク領域の内容が仕様どおりであることを確認する。

選択した問題は，選択欄の(選)をマークしてください。マークがない場合は，採点されません。

問6　プロジェクトの要員計画に関する次の記述を読んで，設問1，2に答えよ。

A社では，社内で使用する新システムの構築に関するプロジェクトの検討を進めてきた。プロジェクトは，システム開発が大規模になるので，要員の確保と納期遵守の観点から，一期開発と二期開発から成る2段階の開発とした。開発チームは，A社の要員と，協力会社B社の要員で編成することにした。

〔プロジェクトの説明〕

(1)　開発規模は，一期開発及び二期開発ともに同一である。

(2)　開発は，ウォータフォールモデルに基づいて，外部設計，内部設計，プログラム開発（単体テストを含む），結合テスト及び総合テストの五つの工程に分ける。

(3)　一期開発は2017年4月に開始する。二期開発は，一期開発のプログラム開発の開始月に開始する。

(4)　各工程での月別要員計画の前提条件は，次のとおりである。

①　業務ノウハウ蓄積の観点から，外部設計工程の要員には，A社の要員を80%以上割り当てる。

②　①の条件を満たす最少の人数をA社の要員数とし，全期間を通して一定の人数とする。すなわち，A社の要員には，各月とも，全員が必ず担当する工程があるものとする。

③　プログラム開発工程には，一期開発及び二期開発ともに，B社の要員だけを割り当てる。

④　プログラム開発工程を除き，各月の必要要員については，まずA社の要員を割り当て，A社の要員だけでは不足する場合に，B社の要員を割り当てる。

⑤　一期開発と二期開発の作業が重なる期間については，A社の要員を，一期開発に優先して割り当てる。

⑥　一人が一期開発と二期開発の作業を同一月に行うことはない。

設問1　要員数に関する次の記述中の［　　］に入れる適切な答えを，解答群の中から選べ。

一期開発及び二期開発における各工程の工数及び配分月数を，表1のとおりに設定した。ここで，配分月数とは，あらかじめ各工程に配分した開発期間（月数）である。二期開発は，一期開発における成果が活用できることから，一期開発よりも，工数を削減することを目標とした。

表1　一期開発及び二期開発における各工程の工数及び配分月数

段階	区分	工程				
		外部設計	内部設計	プログラム開発	結合テスト	総合テスト
一期開発	工数（人月）	42.0	70.0	140.0	52.5	42.0
	配分月数（月）	3	3		2	3
二期開発	工数（人月）	38.1	63.0	127.2	48.0	38.1
	配分月数（月）	3	3		2	3

注記　網掛けの部分は表示していない。

一期開発，二期開発それぞれにおける各工程の各月の要員数は次式で算出する。ここで，各月の要員数は，小数点以下を切り上げた整数値とする。

$$\text{各月の要員数（人）} = \frac{\text{各工程の工数（人月）}}{\text{各工程の配分月数（月）}}$$

A 社の要員数は，〔プロジェクトの説明〕(4) の前提条件に基づき［　a　］人になる。一期開発の外部設計工程では，B 社の要員数は，各月ともに［　b　］人になる。一期開発及び二期開発の全工程について，要員数を求めた結果，各月で B 社の要員数がばらついていることが分かった。そこで，各月の B 社の要員数をできるだけ平準化するために，プログラム開発工程の配分月数を調整することにした。ここで，納期などの制約から，プログラム開発工程で許容される配分月数は，3～5である。

プログラム開発工程の配分月数を一期開発及び二期開発ともに4にする開発ス

ケジュール案を表 2 に示す。この案の場合，2018 年［ c ］が，〔プロジェクトの説明〕(4) の条件② を満たせないことが分かった。これは，プログラム開発工程の配分月数を，“［ d ］”又は“一期開発を 5 に，二期開発を 4 にする”ことで解消できる。

表 2　プログラム開発工程の配分月数を一期開発及び二期開発ともに 4 にする開発スケジュール案

年			2017									2018											
月			4	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12
工程	一期開発		外部設計			内部設計			プログラム開発				結合テスト		総合テスト								
	二期開発								外部設計			内部設計			プログラム開発				結合テスト		総合テスト		
要員数（人）	一期開発	A 社							0	0	0	0											
		B 社							35	35	35	35											
	二期開発	A 社													0	0	0	0					
		B 社													32	32	32	32					

注記　網掛けの部分は表示していない。

a，b に関する解答群

ア　2　　イ　3　　ウ　11　　エ　12

オ　14　　カ　42

c に関する解答群

ア　5 月　　イ　6 月　　ウ　7 月　　エ　8 月

d に関する解答群

ア　一期開発を 3 に，二期開発を 4 にする

イ　一期開発を 3 に，二期開発を 5 にする

ウ　一期開発を 4 に，二期開発を 3 にする

エ　一期開発を 4 に，二期開発を 5 にする

設問 2　要員計画に関する次の記述中の［　　］に入れる適切な答えを，解答群の中から選べ。

E
後

プログラム開発工程の配分月数を，“一期開発を 5 に，二期開発を 4 にする”案を選択した。その上で，更に要員計画策定中にユーザ部門から二期開発の追加要件を受けて見直した開発スケジュール案を表 3 に示す。

表 3　見直した開発スケジュール案

年			2017									2018											
月			4	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12
工程	一期開発		外部設計			内部設計			プログラム開発					結合テスト		総合テスト							
	二期開発								外部設計			内部設計			プログラム開発				結合テスト		総合テスト		
要員数（人）	一期開発	A 社							0	0	0	0	0										
		B 社							28	28	28	28	28	15	15	2	2	2					
	二期開発	A 社												0	0	0	0	0					
		B 社												22	33	33	33	33					

注記　網掛けの部分は表示していない。

各月の B 社の総要員数（B 社の一期開発及び二期開発の要員数の合計）は，ピーク時の 2018 年 4 月に他の月よりも突出する。ピーク時の B 社の総要員数を減らし，3 月から 5 月の 3 か月間の各月の B 社の総要員数を等しくしたい。そのためには，この 3 か月間に実施する各工程の月ごとの要員数は一定ではなくなるが，4 月の一期開発の要員のうち［　e　］人を 3 月に移動し，二期開発の要員のうち［　f　］人を 5 月に移動すればよい。

e，f に関する解答群

ア　1　　イ　2　　ウ　3　　エ　4

オ　5　　カ　6

選択した問題は，選択欄の(選)をマークしてください。マークがない場合は，採点されません。

問7　在庫補充方法の変更に関する次の記述を読んで，設問1～3に答えよ。

食品メーカのM社では，コスト削減やキャッシュフローの改善を目的として，自社の在庫補充方法を変更して在庫量の削減を図ることにした。従来は，毎週月曜日に，保管を委託している全ての配送倉庫の全商品について，配送倉庫ごとに，当該週を含む4週間分の出荷予想量と在庫量を比較し，出荷予想量に満たない量を工場から補充していた。このたびM社では，出荷予想量を算出する期間を見直し，配送倉庫，商品群，賞味期限ごとに在庫基準量を設定し，在庫量が在庫基準量に満たない場合，その差を在庫補充量として補充することにした。この新しい在庫補充方法を次に示す。

〔新しい在庫補充方法〕

(1) 販売量が多い商品群A，中程度の商品群B，少ない商品群Cの商品群別に管理する。

(2) 賞味期限が短い商品（以下，短期品という）は，全ての商品群について2週間分の出荷予想量を在庫基準量とする。

(3) 賞味期限が中程度の商品（以下，中期品という）は，商品群Aについて2週間分の出荷予想量を，商品群B及びCについて3週間分の出荷予想量を在庫基準量とする。

(4) 賞味期限が長い商品（以下，長期品という）は，全ての商品群について3週間分の出荷予想量を在庫基準量とする。

(5) 配送倉庫の中には，工場から遠く，すぐに在庫を補充できない配送倉庫（以下，遠隔地倉庫という）がある。遠隔地倉庫における商品群Aの商品の在庫基準量は，(2)～(4)にかかわらず次のとおりとする。

① 短期品は，2週間分の出荷予想量を在庫基準量とする。

② 中期品は，3週間分の出荷予想量を在庫基準量とする。

③ 長期品は，4週間分の出荷予想量を在庫基準量とする。

(6) 遠隔地倉庫における商品群 B 及び C の商品は全て，(2)～(4) にかかわらず 4 週間分の出荷予想量を在庫基準量とする。

設問 1　在庫量を削減することによって期待される効果に関する次の記述中の［　　］に入れる適切な答えを，解答群の中から選べ。

M 社では，在庫量を削減することによって，原材料購入から商品販売までの期間を短くできるので［ a ］できる。また，［ b ］の削減も期待できる。さらに，在庫として保有する期間が短くなるので，商品の［ c ］についても期待できる。

a に関する解答群

ア　売掛金の回収期間を短縮
イ　買掛金の支払期間を延長
ウ　キャッシュの回収期間を短縮
エ　資金の調達期間を延長

b に関する解答群

ア　在庫スペースの縮小による保管コスト
イ　商品種類の減少による管理コスト
ウ　商品種類の減少による商品開発コスト
エ　フリーキャッシュフローの減少による事務コスト

c に関する解答群

ア　欠品によって販売機会を逸失するリスクの低減
イ　限界利益率の低下
ウ　賞味期限切れが発生するリスクの低減
エ　損益分岐点販売量の増加

設問 2　M 社では，新しい在庫補充方法に合わせて在庫管理システムの変更を検討している。その準備として，在庫基準量を決定するために表 1 に示す決定表を作成した。表 1 中の［　　］に入れる正しい答えを，解答群の中から選べ。

表１　新しい在庫補充方法における在庫基準量の決定表

遠隔地倉庫	N	N	N	N	N	N	Y	Y	Y	Y
A商品群	Y	Y	Y	N	N	N	Y	Y	Y	N
d	Y	N	N	Y	N	N	Y	N	N	−
	−	Y	N	−	Y	N	−	Y	N	−
4週間分の出荷予想量を在庫基準量とする	−	−	−	e				−	−	X
3週間分の出荷予想量を在庫基準量とする	X	−	−					X	−	−
2週間分の出荷予想量を在庫基準量とする	−	X	X					−	X	−

dに関する解答群

ア

短期品
中期品

イ

短期品
長期品

ウ

中期品
短期品

エ

中期品
長期品

オ

長期品
短期品

カ

長期品
中期品

eに関する解答群

ア

X	−	−	X
−	X	X	−
−	−	−	−

イ

X	−	−	X
−	X	−	−
−	−	X	−

ウ

−	−	−	X
X	X	X	−
−	−	−	−

エ

−	−	−	X
X	X	−	−
−	−	X	−

オ

−	−	−	−
X	X	−	X
−	−	X	−

カ

−	−	−	−
X	−	−	X
−	X	X	−

設問３　Ｍ社では，新しい在庫補充方法による在庫量の削減効果を，遠隔地倉庫ではないＮ配送倉庫について見積もることにした。表２は，ある月曜日におけるＮ配送倉庫の在庫量及び４週間分の出荷予想量である。Ｎ配送倉庫の在庫補充量及び在庫量に関する次の記述中の［　　］に入れる適切な答えを，解答群の中から選べ。

表２　Ｎ配送倉庫の在庫量及び４週間分の出荷予想量

単位：百箱

商品	商品群	賞味期限の区分	在庫量	出荷予想量			
				１週目（当該週）	２週目	３週目	４週目
P	A	短期品	120	100	80	50	70
Q	A	中期品	200	120	140	100	120
R	B	中期品	30	30	40	40	30
S	B	長期品	10	20	20	20	20
T	C	長期品	10	10	5	5	10

新しい在庫補充方法に基づく商品Ｐの在庫補充量は［ f ］百箱となる。また，商品Ｒの在庫補充量は，従来の在庫補充方法に基づく在庫補充量に比べて［ g ］百箱の削減となる。商品Ｐ，Ｑ，Ｒ，Ｓ及びＴの合計では，新しい在庫補充方法の導入によって従来の在庫補充方法に比べて，補充後の在庫量を［ h ］百箱削減できる。

f～hに関する解答群

ア　30　　イ　60　　ウ　70　　エ　120　　オ　170

カ　180　　キ　230　　ク　300　　ケ　400　　コ　440

次の問８は必須問題です。必ず解答してください。

問８　次のプログラムの説明及びプログラムを読んで，設問１，２に答えよ。

副プログラム ShortestPath は，N 個（N＞1）の地点と，地点間を直接結ぶ経路及び距離が与えられたとき，出発地から目的地に至る最短経路とその距離を求めるプログラムである。最短経路とは，ある地点から別の地点へ最短距離で移動する際の経由地を結んだ経路である。副プログラム ShortestPath では，出発地の隣接地点から開始して，目的地に向かって最短距離を順次確定する。ある地点の隣接地点とは，その地点から他の地点を経由せずに直接移動できる地点のことである。

図１は，地点数 N が７の経路の例で，経路をグラフで表現したものである。図１において，丸は地点を示し，各地点には０から始まる番号（以下，地点番号という）が順番に割り当てられている。線分は地点間を直接結ぶ経路を示し，線分の上に示す数字はその距離を表す。また，経路上は，双方向に移動できる。

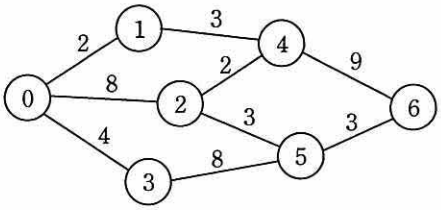


図１　地点数 N が７の経路の例

〔プログラムの説明〕

(1)　副プログラム ShortestPath の引数の仕様を，表１に示す。ここで，出発地から目的地までを結ぶ経路は，少なくとも一つ存在するものとする。また，配列の要素番号は，０から始まる。

表1　副プログラム ShortestPath の引数の仕様

引数	データ型	入出力	説明
Distance[][]	整数型	入力	地点間の距離が格納されている2次元配列
nPoint	整数型	入力	地点数
sp	整数型	入力	出発地の地点番号
dp	整数型	入力	目的地の地点番号
sRoute[]	整数型	出力	出発地から目的地までの最短経路上の地点の地点番号を目的地から出発地までの順に設定する1次元配列
sDist	整数型	出力	出発地から目的地までの最短距離

(2)　Distance[i][j]（i＝0,…,nPoint－1; j＝0,…,nPoint－1）には，地点 i から地点 j までの距離が格納されている。ただし，地点 i と地点 j が同一の地点の場合は 0，地点 i が地点 j の隣接地点ではない場合は －1 が格納されている。図1の例における配列 Distance の内容は，表2のとおりである。

表2　図1の例における配列 Distance の内容

i＼j	0	1	2	3	4	5	6
0	0	2	8	4	−1	−1	−1
1	2	0	−1	−1	3	−1	−1
2	8	−1	0	−1	2	3	−1
3	4	−1	−1	0	−1	8	−1
4	−1	3	2	−1	0	−1	9
5	−1	−1	3	8	−1	0	3
6	−1	−1	−1	−1	9	3	0

(3)　行番号5～10では，変数，配列に初期値を格納する。

①　最短距離を返却する変数 sDist に初期値として ∞（最大値を表す定数）を格納し，出発地から目的地までの最短距離が設定されていないことを示す。

②　最短経路を返却する要素数が nPoint である配列 sRoute の全ての要素に初期値として －1 を格納し，最短経路上の地点の地点番号が設定されていないことを示す。

③　出発地から各地点までの最短距離を設定する配列を pDist とする。pDist は 1

次元配列であり，要素数は nPoint である。配列 pDist の全ての要素に初期値として ∞ を格納する。

④ 出発地から各地点までの最短距離が確定しているかどうかを識別するための配列を pFixed とする。pFixed は 1 次元配列であり，要素数は nPoint である。配列 pFixed の全ての要素に初期値として false を格納し，最短距離が未確定であることを示す。最短距離が確定したときに true を設定する。例えば，出発地から地点 i までの最短距離が確定したとき，pFixed[i] は true となり，その最短距離は pDist[i] に設定されている。pFixed[i] が false の場合は，地点 i までの最短距離は未確定であり，pDist[i] の値は最短距離として確定されていない。

(4) 行番号 11 では，出発地から出発地自体への最短距離 pDist[sp]に 0 を設定する。

(5) 行番号 12 ～ 39 の最短経路探索処理では，出発地から各地点までの最短距離を算出しながら，最短経路を求める。

① 行番号 13 ～ 22

配列 pFixed を調べ，出発地から全ての地点までの最短距離が確定していれば，最短経路探索処理を抜けて (6) に進む。

② 行番号 23 ～ 29

出発地からの最短距離が未確定の地点の中で，出発地からの距離が最も短い地点を探し，その地点を sPoint とし，その地点の最短距離を確定する。

③ 行番号 30 ～ 38

各地点に対して (ア)，(イ) を実行し，①に戻る。

(ア) 地点 sPoint の隣接地点であり，かつ，出発地からの最短距離が未確定であるかどうかを調べる。

(イ) (ア) の条件を満たす地点 j に関して，出発地から地点 sPoint を経由して地点 j に到達する経路の距離を求め，その距離が既に算出してある pDist[j]よりも短ければ，pDist[j]及び pRoute[j]を更新する。ここで，pDist[j]は，出発地から地点 j までの仮の最短距離となる。pRoute[j]には，そのときの，地点 j の直前の経由地の地点番号を設定する。

(6) 行番号 40 ～ 48 では，出発地から目的地までの最短距離を sDist に，最短経路上の地点の地点番号を目的地から出発地までの順に配列 sRoute に設定する。

〔プログラム〕

（行番号）

```
 1  ○副プログラム: ShortestPath( 整数型 : Distance[][], 整数型 : nPoint,
                    整数型 : sp, 整数型 : dp, 整数型 : sRoute[], 整数型 : sDist )
 2  ○整数型 : pDist[nPoint], pRoute[nPoint]
 3  ○論理型 : pFixed[nPoint]
 4  ○整数型 : sPoint, i, j, newDist

 5  ・sDist ← ∞     /* 出発地から目的地までの最短距離に初期値を格納する */

 6  ■ i: 0, i < nPoint, 1
 7  |  ・sRoute[i] ← −1  /* 最短経路上の地点の地点番号に初期値を格納する */
 8  |  ・pDist[i] ← ∞  /* 出発地から各地点までの最短距離に初期値を格納する */
 9  |  ・pFixed[i] ← false  /* 各地点の最短距離の確定状態に初期値を格納する */
10  ■

11  ・pDist[sp] ← 0    /* 出発地から出発地自体への最短距離に 0 を設定する */
12  ■ true  /* 最短経路探索処理 */
13  |  ・i ← 0
14  |  ■ i < nPoint            /* 未確定の地点を一つ探す */
15  |  |  ▲ not(pFixed[i])
16  |  |  |  ・break            /* 最内側の繰返しから抜ける */
17  |  |  ▼
18  |  |  ・i ← i + 1
19  |  ■
20  |  ▲ i = nPoint        /* 出発地から全ての地点までの最短距離が確定 */
21  |  |  ・break           /* していれば，最短経路探索処理を抜ける */
22  |  ▼
23  |  ■ j: i + 1, j < nPoint, 1  /* 最短距離がより短い地点を探す */
24  |  |  ▲ [ a ] and pDist[j] < pDist[i]
25  |  |  |  ・i ← j
26  |  |  ▼
27  |  ■
28  |  ・sPoint ← i  ←――――――――――――――― α
29  |  ・pFixed[ [ b ] ] ← true  /* 出発地からの最短距離を確定する */
30  |  ■ j: 0, j < nPoint, 1
31  |  |  ▲ Distance[sPoint][j] > 0 and not(pFixed[j])
32  |  |  |  ・newDist ← pDist[sPoint] + Distance[sPoint][j]
33  |  |  |  ▲ newDist < pDist[j]
34  |  |  |  |  ・pDist[j] ← newDist
35  |  |  |  |  ・pRoute[j] ← sPoint
36  |  |  |  ▼
37  |  |  ▼
38  |  ■
39  ■  ←――――――――――――――――――――― β
```

```
40  ・sDist ← pDist[dp]
41  ・j ← 0
42  ・i ← dp
43  ■ i ≠ sp
44  │ ・[   c   ] ← i
45  │ ・i ← [   d   ]
46  │ ・j ← j + 1
47  ■
48  ・[   c   ] ← sp
```

設問1　プログラム中の［　　］に入れる正しい答えを，解答群の中から選べ。

a に関する解答群

ア　not(pFixed[i])　　イ　not(pFixed[j])

ウ　pFixed[i]　　エ　pFixed[j]

b に関する解答群

ア　dp　　イ　nPoint　　ウ　nPoint − 1

エ　sp　　オ　sPoint

c，d に関する解答群

ア　pRoute[dp]　　イ　pRoute[i]　　ウ　pRoute[j]　　エ　pRoute[sp]

オ　sRoute[dp]　　カ　sRoute[i]　　キ　sRoute[j]　　ク　sRoute[sp]

設問2　次の記述中の［　　］に入れる正しい答えを，解答群の中から選べ。

図1において，出発地の地点番号 sp の値が 0，目的地の地点番号 dp の値が 6 の場合について，プログラムの動きを追跡する。行番号 12 ～ 39 の最短経路探索処理の繰返しで，行番号 28 の α において sPoint に代入された値は，繰返しの 1 回目は 0，2 回目は 1，3 回目は［　e　］となる。また，行番号 30 ～ 38 の処理が終了した直後の β における配列 pDist と配列 pRoute の値を，表3に示す。最短経路探索処理の繰返しが 3 回目のときの β における配列 pDist の値は［　f　］となり，配列 pRoute の値は［　g　］となる。ここで，配列

pRoute の全ての要素には初期値として 0 が格納されているものとする。

表3　βにおける配列 pDist と配列 pRoute の値

最短経路探索処理の繰返し	配列 pDist	配列 pRoute
1回目	0,2,8,4,∞,∞,∞	0,0,0,0,0,0,0
2回目	0,2,8,4,5,∞,∞	0,0,0,0,1,0,0
3回目	f	g
⋮	⋮	⋮

e に関する解答群

ア　2　　イ　3　　ウ　4　　エ　5　　オ　6

f に関する解答群

ア　0,2,8,4,5,10,14

イ　0,2,8,4,5,10,∞

ウ　0,2,8,4,5,11,14

エ　0,2,8,4,5,11,∞

オ　0,2,8,4,5,12,14

カ　0,2,8,4,5,12,∞

g に関する解答群

ア　0,0,0,0,1,3,0

イ　0,0,0,0,1,3,5

ウ　0,0,0,0,2,2,0

エ　0,0,0,0,2,2,5

オ　0,0,4,0,1,2,0

カ　0,0,4,0,2,2,5

次の問9から問13までの5問については，この中から1問を選択し，選択した問題については，答案用紙の選択欄の(選)をマークして解答してください。

なお，2問以上マークした場合には，はじめの1問について採点します。

問9　次のCプログラムの説明及びプログラムを読んで，設問1～3に答えよ。

マーク式試験の答案を採点するプログラムである。問題と受験者の解答の例を，図1に示す。

問題	受験者の解答
問11　次のうち，面積が最も大きいのはどれか。 ⓐ 2,000mm^2　ⓑ 30cm^2　ⓒ 0.004m^2 問12　$x^2 - x - 2 = 0$ を満たす x の値を，二つ選べ。 ⓐ -2　ⓑ -1　ⓒ 0　ⓓ 1　ⓔ 2	問11:　ⓐ ⓑ ● 問12:　ⓐ ● ⓒ ● ⓔ

図1　問題と受験者の解答の例

ここでは，unsigned int 型のデータは，16ビットのビット列で表し，連続する8ビットが全て0のときは0…0，全て1のときは1…1と表記する。

〔プログラムの説明〕

(1)　関数markingは，1回の実行で受験者1人分の答案を採点する。

(2)　関数 marking の引数の仕様は，次のとおりである。ここで，引数の値に誤りはないものとする。

引数	データ型	入出力	説明
numQ	int	入力	問の数を格納した変数
type[]	int	入力	各問の解答形式を格納した配列
ansC[]	unsigned int	入力	各問の正答を格納した配列
ansE[]	unsigned int	入力	各問の受験者の解答を格納した配列
mark[]	int	出力	各問の採点結果を格納する配列

①　問の数は，numQ である。引数である各配列の要素数も，numQ である。

② 1問当たりの選択肢の個数は16個以下である。

③ type[]の各要素には，答えを一つマークする問の場合は1，複数個マークする問の場合はマークする個数n（2≦n≦8）が格納されている。

④ ansC[]の各要素は，16ビットのビット列で，図2の例に示すように，最上位のビットから1ビットずつ順に選択肢ⓐ，ⓑ，… に対応し，正答の選択肢に対応するビット位置に1が格納されている。他の全てのビットは0である。

⑤ ansE[]の各要素は，ansC[]と同様の形式で，受験者の解答が格納されている。

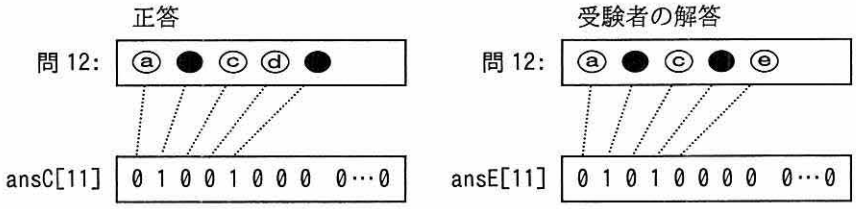


図2　正答及び受験者の解答の格納形式と格納例

(3) 関数 marking は，各問（要素番号 i：0，1，…，numQ−1）について，表1に示す解答形式に応じた採点方法で採点する。採点結果は mark[i] に格納する。

表1　解答形式に応じた各問の採点方法

解答形式	type[i] の値	採点方法及び mark[i] に格納する値
答えを一つマークする問	1	ansC[i] = ansE[i]（正解）なら1， ansC[i] ≠ ansE[i]（不正解）なら0
答えを複数個マークする問	n（2≦n≦8）	ansE[i] 中の1のビットの個数がn＋1個以上なら0，n個以下なら，ansC[i] と ansE[i] の対応するビットがともに1である個数

(4) 関数 marking から呼ばれる関数 countMarkBits は，引数の値に含まれる1のビットの個数を返却値として返す。

〔プログラム1〕

```
void marking(int numQ, int type[], unsigned int ansC[],
             unsigned int ansE[], int mark[]) {
   int i;
   /* α */

   for (i = 0; i < numQ; i++) {
      mark[i] = 0;
      if (type[i] == 1) {
         if (ansE[i] == ansC[i]) {
            mark[i] = 1;
         }
      }
      else if ((2 <= type[i]) && (type[i] <= 8)) {
         if (countMarkBits(  a  ) <= [  b  ]) {
            mark[i] = countMarkBits(  c  );
         }
      }
      /* β */
   }
}
```

設問1　プログラム1中の［　　］に入れる正しい答えを，解答群の中から選べ。

a～cに関する解答群

ア　2	イ　3
ウ　ansC[i]	エ　ansC[i] & ansE[i]
オ　ansC[i] | ansE[i]	カ　ansE[i]
キ　type[i]	ク　type[i] + 1

設問2　関数 countMarkBits のプログラムを，プログラム 2 に示す。次の記述中の［　　］に入れる正しい答えを，解答群の中から選べ。

〔プログラム2〕

```
int countMarkBits(unsigned int ans) {
    int count = 0;
    unsigned int work = ans;

    while (work != 0) {
        count++;
        work = work & (work - 1);    /* γ */
    }
    return count;
}
```

コメント /* γ */ を付した行の代入式を実行するとき，実行前の work の値が 0101 0000 0…0 であれば，実行後の work の値は［ d ］となる。

コメント /* γ */ を付した行の代入式を，

```
work = work && (work - 1);
```

と変更して，プログラム2の実行を試みた場合，［ e ］。

dに関する解答群

ア　0001 0000 0…0　　イ　0100 0000 0…0

ウ　0100 1111 1…1　　エ　1001 0000 0…0

eに関する解答群

ア　演算子&&と&とは実行する演算の内容が同じなので，正しい結果を返す

イ　演算子&&はビットごとの論理積を求めることはしないので，誤った結果を返す

ウ　変数 work には実行前の値が保持されたまま変わらないので，無限ループになる

設問３　次の記述中の［　　］に入れる正しい答えを，解答群の中から選べ。

解答形式に，順不同形式を追加する。順不同形式とは，答えを一つマークする問が２問以上連続していて，共通の解答群から答えを選んでいくが，その選択の順序は入れ替わってもよい形式である。図３の例では，問 21，22，23 の順に正答ⓑ，ⓓ，ⓔが設定されているが，これを受験者の解答の例のように異なる順序で解答してもよい。正しい答えが幾つ選択されたかによって，採点結果が決まる。

問題	
問 21～23　次の府県のうち，海に面していないのは［問 21］，［問 22］及び［問 23］である。 ⓐ 大阪　ⓑ 岐阜　ⓒ 京都　ⓓ 滋賀　ⓔ 奈良　ⓕ 兵庫　ⓖ 三重	
正答	受験者の解答
問 21:　ⓐ ● ⓒ ⓓ ⓔ ⓕ ⓖ 問 22:　ⓐ ⓑ ⓒ ● ⓔ ⓕ ⓖ 問 23:　ⓐ ⓑ ⓒ ⓓ ● ⓕ ⓖ	問 21:　ⓐ ⓑ ⓒ ⓓ ● ⓕ ⓖ 問 22:　ⓐ ● ⓒ ⓓ ⓔ ⓕ ⓖ 問 23:　ⓐ ⓑ ⓒ ● ⓔ ⓕ ⓖ

図３　順不同形式の問題，正答及び受験者の解答の例

順不同形式の問題に対する解答を含む答案を採点するために，プログラム１を次のとおり修正した。

(1)　プログラム１中のコメント /* α */ の行を，次の１行で置き換える。

```
unsigned int sumC, sumE;
```

(2)　プログラム１中のコメント /* β */ の行を，次の 13 行で置き換える。

```
else if ((11 <= type[i]) && (type[i] <= 13)) {
    if (type[i] == 11) {
        sumC = 0;
        sumE = 0;
    }
    sumC = sumC | ansC[i];
    if (countMarkBits(ansE[i]) == 1) {
        sumE = sumE | ansE[i];
    }
    if (type[i] == 13) {
        mark[i] = countMarkBits(sumC & sumE);
    }
}
```

順不同形式の問を表す type[] の値は，11 ～ 13 である。一組の順不同形式の問題において，値 11 は先頭の問を表し，値 13 は末尾の問を表す。値 12 は 3 問以上から成る順不同形式の問題における中間の問を表す。

図 3 の順不同形式の問題に対する解答を含む答案を，修正したプログラム 1 で採点する。ある受験者の解答の採点例として，関数 marking に渡された各配列要素の内容，及びこれらの内容に基づく採点結果を，図 4 に示す。

解答形式		正答		受験者の解答		採点結果	
type[20]	11	ansC[20]	0100 0000 0…0	ansE[20]	0000 1000 0…0	mark[20]	f
type[21]	12	ansC[21]	0001 0000 0…0	ansE[21]	0000 1000 0…0	mark[21]	
type[22]	13	ansC[22]	0000 1000 0…0	ansE[22]	0101 0000 0…0	mark[22]	

図 4　関数 marking に渡された配列の内容及び採点結果

また，図 4 に示す解答の採点が完了した時点で，変数 sumE には値 [g] が格納されている。

fに関する解答群

ア	イ	ウ	エ	オ	カ
0	0	0	0	1	1
0	0	1	1	0	0
1	3	0	2	0	2

gに関する解答群

ア　0…0 0…0　　イ　0000 1000 0…0

ウ　0101 0000 0…0　　エ　0101 1000 0…0

選択した問題は，選択欄の(選)をマークしてください。マークがない場合は，採点されません。

問10　次のCOBOLプログラムの説明及びプログラムを読んで，設問1，2に答えよ。

〔プログラムの説明〕

八つの支店をもつZ社は4月～翌年3月を事業年度としており，毎年4月に前年度の売上データを集計して売上マスタファイルに格納する。このプログラムは，売上伝票ファイルに格納された前年度の売上データを集計して，売上マスタファイルに格納する。

(1)　売上伝票ファイルは，図1に示すレコード様式の順ファイルである。

支店番号 2桁	伝票番号 6桁	売上日 8桁	売上金額 6桁

図1　売上伝票ファイルのレコード様式

①　売上伝票ファイルには，前年度の全支店の売上データが，順不同で格納されている。レコード件数がゼロの月はない。

②　支店番号には，各支店に対応する01～08の番号が格納されている。支店1は01，支店2は02，…，支店8は08である。

③　伝票番号には，各支店で一意となる6桁の数字が格納されている。

④　売上日には，西暦の年，月，日が，それぞれ4桁，2桁，2桁で格納されている。

(2)　売上マスタファイルは，図2に示すレコード様式の索引ファイルである。主キーは売上年月である。

売上年月 6桁	売上金額合計 10桁	支店別売上金額			
		支店1 9桁	支店2 9桁	…	支店8 9桁

図2　売上マスタファイルのレコード様式

①　売上年月には，西暦の年，月が，それぞれ4桁，2桁で格納される。

COBOL

② 支店別売上金額には，当該年月における各支店の売上金額が格納される。各支店の月間の売上金額が9桁を超えることはない。

③ 売上金額合計には，当該年月における全支店の売上金額の合計が格納される。

④ 売上マスタファイルには，このプログラムを実行する時点で，2000 年度から前々年度までのデータが格納されている。

〔プログラム〕

（行番号）

```
 1  DATA DIVISION.
 2  FILE SECTION.
 3  SD SRT-FILE.
 4  01 SRT-REC.
 5     02 SRT-BRANCH     PIC 9(2).
 6     02 SRT-NO         PIC 9(6).
 7     02 SRT-DATE.
 8        03 SRT-YYYYMM     PIC 9(6).
 9        03 SRT-DD         PIC 9(2).
10     02 SRT-AMOUNT     PIC 9(6).
11  FD SAL-FILE.
12  01 SAL-REC        PIC X(22).
13  FD MST-FILE.
14  01 MST-REC.
15     02 MST-YYYYMM.
16        03 MST-YYYY       PIC 9(4).
17        03 MST-MM         PIC 9(2).
18     02 MST-AMOUNT     PIC 9(10).
19     02 MST-TABLE.
20        03 MST-BRANCH     PIC 9(9) OCCURS 8.
21  WORKING-STORAGE SECTION.
22  77 CNT          PIC 9(1).
23  77 SRT-FLAG     PIC X(1) VALUE SPACE.
24     88 SRT-EOF       VALUE "E".
25  PROCEDURE DIVISION.
26  MAIN-PROC.
27      OPEN EXTEND MST-FILE.
28      SORT SRT-FILE ASCENDING KEY [      a      ]
29         USING SAL-FILE
30         OUTPUT PROCEDURE IS RET-PROC.
31      CLOSE MST-FILE.
32      STOP RUN.
```

COBOL

```
33  RET-PROC.
34      INITIALIZE MST-REC.
35      PERFORM UNTIL SRT-EOF
36         RETURN SRT-FILE AT END     SET SRT-EOF TO TRUE
37                                    PERFORM WRI-PROC
38                          NOT AT END PERFORM ADD-PROC
39         END-RETURN
40      END-PERFORM.
41  ADD-PROC.
42      IF [          b          ] THEN
43         PERFORM WRI-PROC
44         INITIALIZE MST-REC
45         MOVE SRT-YYYYMM TO MST-YYYYMM
46      END-IF.
47      [          c          ].
48  WRI-PROC.
49      IF MST-YYYY NOT = ZERO THEN
50         PERFORM VARYING CNT FROM 1 BY 1 UNTIL CNT > 8
51            [          d          ]
52         END-PERFORM
53         WRITE MST-REC
54      END-IF.
```

COBOL

設問1　プログラム中の[　　]に入れる正しい答えを，解答群の中から選べ。

aに関する解答群

ア　SRT-AMOUNT　　イ　SRT-BRANCH　　ウ　SRT-DATE　　エ　SRT-NO

bに関する解答群

ア　MST-YYYYMM = SRT-YYYYMM　　イ　MST-YYYYMM NOT = SRT-YYYYMM

ウ　MST-YYYYMM = ZERO　　エ　SRT-YYYYMM = ZERO

c，dに関する解答群

ア　ADD MST-BRANCH(CNT) TO MST-AMOUNT

イ　ADD MST-BRANCH(SRT-BRANCH) TO MST-AMOUNT

ウ　ADD SRT-AMOUNT TO MST-BRANCH(CNT)

エ　ADD SRT-AMOUNT TO MST-BRANCH(SRT-BRANCH)

オ　MOVE SRT-AMOUNT TO MST-BRANCH(SRT-BRANCH)

設問2　前年度の売上データを集計して，売上マスタファイルに格納する際に，第1四半期から第4四半期の順に，当該四半期における全支店の売上金額の合計を表示するようにプログラムを変更する。第1四半期から第4四半期は，それぞれ4月～6月，7月～9月，10月～12月，1月～3月である。表示例を図3に示す。表1中の［　　］に入れる正しい答えを，解答群の中から選べ。

```
1ST QUARTER:     736,055,716
2ND QUARTER:   1,088,207,339
3RD QUARTER:     865,163,702
4TH QUARTER:   1,224,859,224
```

図3　表示例

表1　プログラムの変更内容

処置	変更内容
行番号17と18の間に追加	88 QUART-MONTH VALUES ［　f　］.
行番号24と25の間に追加	77 QUART-AMOUNT PIC 9(11) VALUE ZERO. 77 QUART-AMOUNT-D PIC ZZ,ZZZ,ZZZ,ZZ9. 77 CR-QUART PIC 9(1) VALUE 1. 01 QUART-STR PIC X(8) VALUE "STNDRDTH". 01 QUART-TBL REDEFINES QUART-STR. 02 QUART-NO PIC X(2) OCCURS 4.
［　e　］に追加	ADD MST-AMOUNT TO QUART-AMOUNT IF QUART-MONTH THEN MOVE QUART-AMOUNT TO QUART-AMOUNT-D DISPLAY CR-QUART QUART-NO(CR-QUART) " QUARTER: " QUART-AMOUNT-D MOVE ZERO TO QUART-AMOUNT ［　g　］ END-IF

eに関する解答群

ア　行番号30と31の間　　イ　行番号40と41の間

ウ　行番号47と48の間　　エ　行番号52と53の間

fに関する解答群

ア　1 THRU 12　　イ　4 THRU 3　　ウ　4, 7, 10, 1　　エ　6, 9, 12, 3

gに関する解答群

ア　ADD 1 TO CR-QUART　　イ　ADD 3 TO CR-QUART

ウ　MOVE CNT TO CR-QUART　　エ　MOVE ZERO TO CR-QUART

選択した問題は，選択欄の(選)をマークしてください。マークがない場合は，採点されません。

問11　次のJavaプログラムの説明及びプログラムを読んで，設問1，2に答えよ。

〔プログラムの説明〕

電気料金プラン（以下，プランという）を比較するためのプログラムである。電気料金は，基本料金，及び電気の使用量（以下，電力量という）から算出される電力量料金から成る。電力量の単位は，キロワット時（kWh）である。電力量料金を算出するための1kWh当たりの料金（以下，料金単価という）は，電力量に応じて段階的に変わる。プランA及びプランBの料金表を表1に示す。

表1　プランA及びプランBの料金表

プラン		プランA	プランB
基本料金（円）		1123.30	1040.10
料金単価（円／kWh）	120kWhまで	19.62	18.17
	120kWh超過300kWhまで	26.10	24.17
	300kWh超過分	30.12	27.77

例えば，プランAで，使用した電力量が200.5kWhのとき，120kWh分に対して19.62円／kWh，残りの80.5kWhに対して26.10円／kWhの料金単価が適用される。

(1)　抽象クラスTierTableは，段階的に変化する値のペアをテーブルとして表現する。値は，型doubleで与えられる。

①　コンストラクタは，可変長の引数で与えられたdoubleの数値2個ずつをペアとして配列にし，更にその配列を要素とする配列を生成し，フィールドpairsに保持する。引数tiersの長さが奇数のときは，IllegalArgumentExceptionを投げる。

②　抽象メソッドmapは，引数で与えられた数値を別の数値に変換して返す。

(2)　クラスTieredRateTableは，料金単価のテーブルを表す。

①　コンストラクタは，引数で与えられた電力量とそれに対応する料金単価のペア

からテーブルを作成する。

② メソッド map は，引数で与えられた電力量から電力量料金を計算し，その値を返す。

(3) クラス RatePlan は，プランを表す。

① コンストラクタは，引数で与えられたプラン名，基本料金，料金単価のテーブルで表されるインスタンスを生成する。

② メソッド getName は，プラン名を返す。

③ メソッド getPrice は，引数で与えられた電力量から電力量料金を計算し，その値と基本料金の合計を型 int の数値で返す。このとき，小数点以下は切り捨てられる。

(4) クラス Main は，表 1 を基にプラン A 及びプラン B を表す RatePlan のインスタンスを生成し，電力量が 543.0 kWh のときの電気料金を比較する。メソッド main を実行すると，図 1 の結果が得られた。

プラン B が 1175 円安い

図 1　メソッド main の実行結果

Java

〔プログラム 1〕

```
[ a ] class TierTable {
   final double[][] pairs;

   TierTable(double... tiers) {
      if (tiers.length % 2 == 1) {
         throw new IllegalArgumentException("不正な長さ: " + tiers.length);
      }
      double[][] a = new double[tiers.length / 2][];
      for (int i = 0; i < tiers.length; i += 2) {
         a[ b ] = new double[] { tiers[i], tiers[i + 1] };
      }
      this.pairs = a;
   }

   abstract double map(double amount);
}
```

〔プログラム2〕

```
class TieredRateTable extends TierTable {
   TieredRateTable(double... tiers) {
      super(tiers);
   }

   double map(double amount) {
      double charge = 0;
      for (int i = 0; i < pairs.length; i++) {
         if (i + 1 < pairs.length && amount > pairs[i + 1][0]) {
            charge += (pairs[   c   ] - pairs[   d   ]) * pairs[i][1];
         } else {
            charge += (amount - pairs[i][0]) * pairs[i][1];
            break;
         }
      }
      return charge;
   }
}
```

〔プログラム3〕

```
class RatePlan {
   private final String name;
   private final double basicCharge;
   private final TierTable pricingTiers;

   RatePlan(String name, double basicCharge, TierTable pricingTiers) {
      this.name = name;
      this.basicCharge = basicCharge;
      this.pricingTiers = pricingTiers;
   }

   String getName() { return name; }

   int getPrice(double amount) {
      return (int) (basicCharge + pricingTiers.map(amount));
   }
}
```

〔プログラム4〕

```
public class Main {
   public static void main(String[] args) {
      RatePlan planA = new RatePlan("プランA", 1123.30,
         new TieredRateTable(0, 19.62, 120, 26.10, 300, 30.12));
      RatePlan planB = new RatePlan("プランB", 1040.10,
         new TieredRateTable(0, 18.17, 120, 24.17, 300, 27.77));

      double amount = 543.0;
      int d = planA.getPrice(amount) - planB.getPrice(amount);
      if (d < 0) {
         System.out.printf("%sが%d円安い%n", planA.getName(), -d);
      } else if (d > 0) {
         System.out.printf("%sが%d円安い%n", planB.getName(), d);
      } else {
         System.out.println("両プランで同額");
      }
   }
}
```

Java

設問1　プログラム中の［　　］に入れる正しい答えを，解答群の中から選べ。

aに関する解答群

ア　abstract	イ　final	ウ　private
エ　protected	オ　public	カ　static

bに関する解答群

ア　i	イ　i - 1	ウ　i * 2
エ　i / 2	オ　i / 2 + 1	カ　i + 1

c，dに関する解答群

ア　[i + 1][0]	イ　[i + 1][1]	ウ　[i + 1][i]
エ　[i][0]	オ　[i][1]	カ　[i][i + 1]

設問2　割引プランは，指定した他のサービスを電気と一緒に利用した場合には，プラン A やプラン B という当初のプランで計算した電気料金を，電気料金の額に応じて割り引くプランである。割引プランの割引率の例を，表2に示す。

表2　割引プランの割引率の例

電気料金	割引率
5,000 円未満	1%
5,000 円以上 8,000 円未満	3%
8,000 円以上	5%

プログラム5は，割引率を求めるためのクラス DiscountTable である。メソッド map は，引数で与えられた電気料金から割引率を求め，その値を返す。ここで，割引率は小数で与えるものとする。例えば，1%は 0.01 である。

〔プログラム5〕

```
class DiscountTable extends TierTable {
   DiscountTable(double... tiers) {
      super(tiers);
   }

   double map(double amount) {
      for (int i = pairs.length - 1; i >= 0; i--) {
         if (amount >= pairs[i][0]) {
            return pairs[i][1];
         }
      }
      throw new IllegalArgumentException("amount = " + amount);
   }
}
```

プログラム 6 は，割引プランを表すためのクラス DiscountPlan である。DiscountPlan は，クラス RatePlan を拡張し，上位クラスである RatePlan のメソッド getPrice で求めた電気料金から割引率を求め，割引を適用した金額を電気料金として計算する。プログラム中の［　　］に入れる正しい答えを，解答群の中から選べ。

〔プログラム6〕

```
class DiscountPlan [   e   ] RatePlan {
    private final TierTable discountTiers;

    DiscountPlan(String name, double basicCharge,
                 TierTable pricingTiers, TierTable discountTiers) {
        super(name, basicCharge, pricingTiers);
        this.discountTiers = discountTiers;
    }

    int getPrice(double amount) {
        int price = [   f   ].getPrice(amount);
        return (int) (price * [   g   ]);
    }
}
```

eに関する解答群

ア extends	イ implements	ウ imports
エ public	オ throws	

fに関する解答群

ア ((RatePlan) this)	イ RatePlan	ウ super
エ this	オ TieredRateTable	カ TierTable

gに関する解答群

ア (1.0 - discountTiers.map(amount))

イ (1.0 - discountTiers.map(basicCharge))

ウ (1.0 - discountTiers.map(price))

エ discountTiers.map(amount)

オ discountTiers.map(basicCharge)

カ discountTiers.map(price)

Java

選択した問題は，選択欄の(選)をマークしてください。マークがない場合は，採点されません。

問 12　次のアセンブラプログラムの説明及びプログラムを読んで，設問 1～3 に答えよ。

〔プログラム 1 の説明〕

64 ビット符号なし整数の加算を行う副プログラム ADD64 である。64 ビット符号なし整数の主記憶への格納方法を，図 1 に示す。

N 番地	N＋1 番地	N＋2 番地	N＋3 番地
最上位 16 ビット	第 2 位 16 ビット	第 3 位 16 ビット	最下位 16 ビット

図 1　64 ビット符号なし整数の主記憶への格納方法

(1)　GR1 及び GR2 には，それぞれ次の内容が設定されて，主プログラムから渡される。

GR1，GR2：加算の対象となる二つの 64 ビット符号なし整数を格納する領域の先頭アドレス。各領域はそれぞれ連続した 4 語から成る。

ここで，GR1 と GR2 の値が等しい場合を除き，GR1 と GR2 が指し示す領域は重ならないものとする。

(2)　副プログラム ADD64 は，GR1 に設定されたアドレスから始まる連続した 4 語の領域に，加算の結果を格納する。このとき，桁あふれは発生しないものとする。

(3)　副プログラム ADD64 から戻るとき，汎用レジスタ GR1～GR7 の内容は元に戻す。

〔プログラム 1〕

(行番号)

```
1  ADD64    START
2           RPUSH
3           LD     GR0,=0
4           LAD    GR3,3,GR1
5           LAD    GR4,3,GR2
```

```
 6  LOOP     LD     GR5,=0
 7           ADDL   GR0,0,GR3
 8           JOV    OV1
 9           JUMP   NOV1
10  OV1      [      a      ]
11  NOV1     ADDL   GR0,0,GR4
12           JOV    OV2
13           JUMP   NOV2
14  OV2      [      a      ]
15  NOV2     ST     GR0,0,GR3
16           LD     GR0,GR5
17           [      b      ]
18           JZE    EXIT
19           SUBL   GR3,=1
20           SUBL   GR4,=1
21           [      c      ]
22  EXIT     RPOP
23           RET
24           END
```

設問１　プログラム１中の［　　］に入れる正しい答えを，解答群の中から選べ。

a，b に関する解答群

ア　AND　GR5,=1　　イ　CPL　GR3,=0　　ウ　CPL　GR3,GR1

エ　CPL　GR3,GR2　　オ　LD　GR5,=1

c に関する解答群

ア　JOV　LOOP　　イ　JUMP　LOOP　　ウ　JZE　LOOP

設問２　次の記述中の［　　］に入れる正しい答えを，解答群の中から選べ。

GR1 及び GR2 に設定したアドレスから始まる連続した 4 語の領域に，図 2 のとおりに値を格納して副プログラム ADD64 を実行したとき，行番号 7 の命令を 2 回目に実行した後の GR0 の値は［　d　］である。

(GR1)番地	(GR1)＋1 番地	(GR1)＋2 番地	(GR1)＋3 番地
#3F1D	#B759	#2E0C	#A684

(GR2)番地	(GR2)＋1 番地	(GR2)＋2 番地	(GR2)＋3 番地
#2E0C	#A684	#3F1D	#B759

図 2　主記憶に格納した値

d に関する解答群

ア　#0000　　イ　#0001　　ウ　#2E0C

エ　#2E0D　　オ　#3F1D　　カ　#3F1E

設問 3　副プログラム ADD64 を使用して，32 ビット符号なし整数の乗算を行う副プログラム MUL32 を作成した。プログラム 2 中の［　　］に入れる正しい答えを，解答群の中から選べ。

〔プログラム 2 の説明〕

(1)　32 ビット符号なし整数の主記憶への格納方法を，図 3 に示す。

N 番地	N＋1 番地
上位 16 ビット	下位 16 ビット

図 3　32 ビット符号なし整数の主記憶への格納方法

(2)　GR1～GR3 には，それぞれ次の内容が設定されて，主プログラムから渡される。

GR1，GR2：乗算の対象となる二つの 32 ビット符号なし整数を格納する領域の先頭アドレス。各領域はそれぞれ連続した 2 語から成る。

GR3：乗算結果となる 64 ビット符号なし整数を格納する領域の先頭アドレス。領域は連続した 4 語から成る。

ここで，GR1，GR2 及び GR3 が指し示す領域は互いに重ならないものとする。

(3)　副プログラム MUL32 から戻るとき，汎用レジスタ GR1～GR7 の内容は元に戻す。

〔プログラム 2〕

```
MUL32     START
          RPUSH
          LAD   GR7,TEMP    ; 初期化
          LD    GR0,0,GR1   ;┐ GR1 から始まる 2 語の領域の値を,
          ST    GR0,2,GR7   ;│ TEMP から始まる 4 語の領域のうちの
          LD    GR0,1,GR1   ;│ 下位 2 語に格納
          ST    GR0,3,GR7   ;┘
          LD    GR0,=0
          ST    GR0,0,GR7   ;┐ TEMP から始まる 4 語の領域のうちの
          ST    GR0,1,GR7   ;┘ 上位 2 語に 0 を格納
          ST    GR0,0,GR3   ;┐
          ST    GR0,1,GR3   ;│ GR3 から始まる 4 語の領域に 0 を格納
          ST    GR0,2,GR3   ;│
          ST    GR0,3,GR3   ;┘
          LD    GR5,=0      ; ループカウンタ
          LD    GR4,GR2     ; GR2 の値を GR4 に退避
LOOP      LD    GR6,GR5
          SUBL  GR6,=16
          [      e      ]
          LD    GR0,0,GR2
          SRL   GR0,0,GR6
          JUMP  TESTBIT
LOWORD    LD    GR0,1,GR2
          SRL   GR0,0,GR5
TESTBIT   AND   GR0,=#0001
          JZE   EXITLOOP
          LD    GR1,GR3
          LAD   GR2,TEMP
          CALL  ADD64
EXITLOOP  CPL   GR5,=31
          JZE   EXIT
          ADDL  GR5,=1
          LAD   GR1,TEMP
          [      f      ]
          CALL  ADD64
          LD    GR2,GR4     ; GR4 に退避した値を GR2 に復帰
          JUMP  LOOP
EXIT      RPOP
          RET
TEMP      DS    4
          END
```

アセンブラ

e，fに関する解答群

ア	JMI	LOWORD	イ	JNZ	LOWORD	ウ	JPL	LOWORD
エ	JZE	LOWORD	オ	LAD	GR2, TEMP	カ	LD	GR2, =1
キ	LD	GR2, GR3	ク	LD	GR2, TEMP			

選択した問題は，選択欄の(選)をマークしてください。マークがない場合は，採点されません。

問 13　次の表計算のワークシート及びマクロの説明を読んで，設問 1～3 に答えよ。

〔表計算の説明〕

T 社の U さんは，3 種類の商品 X，Y，Z を納品書に基づいて梱(こん)包し，宅配便業者に集荷を依頼して発送する業務を担当している。従来，発送する商品をまとめて荷物として梱包した後に，個別に重量を量って料金計算をしていたが，表計算ソフトを用いて作業の効率化を図ることにした。

〔ワークシート：料金表〕

商品の発送に利用する宅配便サービスの発送料金は，料金計算上同一地域とする都道府県の区分である着地域区分と，荷物の重量で区分された重量区分に基づいて決定される。ワークシート“料金表”には，T 社の所在地からの発送料金を検索するために利用する“料金表”が格納されている。ワークシート“料金表”の例を，図 1 に示す。

	A	B	C	D	E	…	M	N
1	料金表（単位：円）							
2	着地域区分		北海道	北東北	南東北	…	九州	沖縄
3	都道府県名		北海道	青森県	宮城県	…	福岡県	沖縄県
4				秋田県	山形県	…	佐賀県	
5				岩手県	福島県	…	長崎県	
6						…	熊本県	
7						…	大分県	
8						…	宮崎県	
9						…	鹿児島県	
10						…		
11	重量区分	1	1,188	864	756	…	1,188	1,296
12		2	1,458	1,134	1,026	…	1,458	1,836
13		3	1,728	1,404	1,296	…	1,728	2,376
14		4	1,998	1,674	1,566	…	1,998	3,186
15		5	2,268	1,944	1,836	…	2,268	3,996
16				1		…		

図 1　ワークシート“料金表”の例

(1) セル C2～N2 には着地域区分の名称が，セル C3～N10 には，それぞれの列の行 2 の着地域区分に属する都道府県名が，上のセルから順に入力されている。全ての都道府県はいずれか一つの着地域区分に必ず属する。

(2) セル B11～B15 には，重量区分（1～5）が入力されている。

(3) セル C11～N15 には，着地域区分と重量区分に対応する発送料金が入力されている。

(4) セル C16～N16 は，料金計算で着地域区分を検索するための作業領域である。

〔ワークシート：商品重量〕

ワークシート“商品重量”には，料金計算で商品の重量を検索するときに利用する“商品重量表”が格納されている。セル A3～A5 には商品名が，セル B3～B5 には各商品 1 個の重量が入力されている。ワークシート“商品重量”を，図 2 に示す。

	A	B
1	商品重量表	
2	商品名	重量（g）
3	X	460
4	Y	720
5	Z	1,200

図 2　ワークシート“商品重量”

〔ワークシート：重量区分〕

ワークシート“重量区分”には，荷物の重量から重量区分を検索するときに利用する“重量区分表”が格納されている。ワークシート“重量区分”を，図 3 に示す。

	A	B	C	D	E
1	重量区分表				
2	重量区分	重量の上限(g)	箱重量(g)	発送可能な商品の総重量の範囲(g)	
3				から	まで
4	1	2,000	135	1	1,865
5	2	5,000	260	1,866	4,740
6	3	10,000	415	4,741	9,585
7	4	20,000	870	9,586	19,130
8	5	30,000	1,320	19,131	28,680

図3　ワークシート“重量区分”

(1)　セル A4～A8 には重量区分が，セル B4～B8 には各重量区分で発送できる荷物の重量の上限が入力されている。この宅配便サービスで発送できる荷物の重量は 30,000 g までである。

(2)　商品の梱包には，重量区分に応じた梱包用の箱を用いる。セル C4～C8 には，各重量区分で用いる梱包用の箱の重量が入力されている。

(3)　セル D4～E8 には，箱の重量を加えたときに，当該重量区分で発送可能な商品の総重量の範囲が入力されている。

〔ワークシート：料金計算〕

ワークシート“料金計算”には，“料金計算表”を作成して格納する。セル A4 に発送先の都道府県名，セル B4～D4 に発送する商品 X，Y 及び Z の数量を入力すると，商品総重量，重量区分，梱包後重量及び発送料金を表示する。ワークシート“料金計算”の例を，図 4 に示す。

	A	B	C	D	E	F	G	H
1	料金計算表							
2	発送先都道府県名	発送商品数量			商品総重量(g)	重量区分	梱包後重量(g)	発送料金(円)
3		X	Y	Z				
4	青森県	12	6	4	14,640	4	15,510	1,674

図4　ワークシート“料金計算”の例

設問1　ワークシート“料金表”に関する次の記述中の［　　］に入れる正しい答えを，解答群の中から選べ。

セル C16 には，セル C3～C10 の中に，ワークシート“料金計算”のセル A4 に入力された発送先都道府県名と一致するものがあるときには1を，そうでなければ空値を表示する次の式を入力し，セル D16～N16 に複写する。

IF(料金計算!$A4 = null, null, ［　a　］)

aに関する解答群

ア　IF(条件付個数(C3:C10, ＝料金計算!$A4)＝0, 1, null)

イ　IF(条件付個数(C3:C10, ＝料金計算!$A4)＝1, 1, null)

ウ　IF(条件付個数(C3:C10, ＝料金計算!$A4)＞1, null, 1)

エ　IF(条件付個数($C3:$C10, ＝料金計算!$A4)＝0, null, 1)

オ　IF(条件付個数($C3:$C10, ＝料金計算!$A4)＝1, null, 1)

カ　IF(条件付個数($C3:$C10, ＝料金計算!$A4)＞1, 1, null)

設問2　ワークシート“料金計算”に関する次の記述中の［　　］に入れる正しい答えを，解答群の中から選べ。

(1)　セル E4 には，発送する商品の総重量を表示する次の式を入力する。

B4＊商品重量!B3＋C4＊商品重量!B4＋D4＊商品重量!B5

(2)　セル F4 には，セル E4 の値が0のときは“－”を，セル E4 の値が発送可能な商品の総重量の上限を超えるときは“×”を表示し，それ以外のときは，セル E4 の商品総重量の商品を一つの荷物として発送できる最も小さい重量区分を，ワークシート“重量区分”から検索して表示する次の式を入力する。

IF(E4＝0, '－', ［　b　］)

(3)　セル G4 には，セル F4 の値が“－”又は“×”のときは“－”を表示し，それ以外のときは，セル E4 の商品総重量にセル F4 の重量区分で利用する箱の重量を加えた値を梱包後重量として表示する式を入力する。

(4)　セル H4 には，セル A4～D4 に入力した条件でワークシート“料金表”か

ら発送料金の検索ができないときは“－”を表示し，それ以外のときは，セルA4の発送先都道府県名とセルF4の重量区分に該当する発送料金をワークシート“料金表”から検索して表示する次の式を入力する。

IF([c] ,'－', [d])

bに関する解答群

ア　IF(E4＞重量区分!E8,'×',照合一致(E4,重量区分!D4:D8,1))

イ　IF(E4＞重量区分!E8,'×',照合一致(E4,重量区分!D4:D8,－1))

ウ　IF(E4＞重量区分!E8,'×',照合一致(E4,重量区分!E4:E8,1))

エ　IF(E4＞重量区分!E8,'×',照合一致(E4,重量区分!E4:E8,－1))

cに関する解答群

ア　論理積(G4≠'－',A4≠null)

イ　論理積(論理和(F4＝'－',F4＝'×'),A4＝null)

ウ　論理和(G4＝'－',合計(料金表!C16:N16)＝0)

エ　論理和(論理積(F4≠'－',F4≠'×'),合計(料金表!C16:N16)＝1)

dに関する解答群

ア　垂直照合(F4,料金表!B11:N15,照合一致(1,料金表!C16:N16,0),0)

イ　垂直照合(照合一致(1,料金表!C16:N16,0),料金表!B11:N15,F4,0)

ウ　表引き(料金表!C11:N15,F4,照合一致(1,料金表!C16:N16,0))

エ　表引き(料金表!C11:N15,照合一致(1,料金表!C16:N16,0),F4)

設問3　ワークシート“料金計算”で，商品総重量が発送可能な商品の総重量の上限を超え，重量区分欄に“×”が表示されると，Uさんは，商品を発送可能な重量に収まるような組合せに分割して梱包し，発送しなければならない。この商品の組合せを求めるために，ワークシート“料金計算”に新たな表を追加するとともに，マクロ Package_count を作成して格納した。マクロ Package_count 中の [　　] に入れる正しい答えを，解答群の中から選べ。

表計算

〔“梱包作業表”及び“梱包指示表”の追加〕

マクロ Package_count を実行すると，商品を分割して発送するときの商品の組合せを“梱包作業表”を利用して求め，その結果が“梱包指示表”に格納される。“梱包作業表”及び“梱包指示表”を追加したワークシート“料金計算”の例を，図5に示す。

	A	B	C	D	E	F	G	H
1	料金計算表							
2	発送先	発送商品数量			商品総重量	重量	梱包後重量	発送料金
3	都道府県名	X	Y	Z	(g)	区分	(g)	(円)
4	青森県	10	36	30	66,520	×	—	—
5								
6	梱包作業表							
7		梱包数量			商品総重量	重量	発送料金	
8		X	Y	Z	(g)	区分	(円)	
9	残数量	0	0	0	0	—	—	
10	作業数量	4	7	2	9,280			
11								
12								
13	梱包指示表							
14	荷物番号	発送商品数量			商品総重量	重量	発送料金	
15		X	Y	Z	(g)	区分	(円)	
16	1	6	1	21	28,680	5	1,944	
17	2	0	28	7	28,560	5	1,944	
18	3	4	7	2	9,280	3	1,404	
19								
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
25								
26	計	10	36	30	66,520		5,292	

図5　“梱包作業表”と“梱包指示表”を追加したワークシート“料金計算”の例

(1)　セル B9～D9 には，まだ梱包の組合せが決定していない商品の数量（以下，残数量という）がマクロ Package_count によって格納される。

(2)　セル E9 には，まだ梱包の組合せが決定していない商品の総重量（以下，残重量という）を表示する式を入力する。

(3)　セル F9 には，セル E9 の値が 0 のときは“－”を，セル E9 の値が発送可能

な商品の総重量の上限を超えるときは 5 を，それ以外のときは，セル E9 の商品総重量の商品を一つの荷物として発送できる最も小さい重量区分を表示する式を入力する。

(4) セル G9 には，セル E9 の値が 0 のときは“－”を表示し，それ以外のときは，セル A4 の発送先都道府県名とセル F9 の重量区分に該当する発送料金を表示する式を入力する。

(5) セル B10～D10 は，マクロ Package_count が作業領域として利用し，それぞれ商品 X，Y 及び Z の数量を格納する。また，セル E10 には，セル B10～D10 に格納された数量の各商品の重量の合計（以下，作業重量という）を表示する式を入力する。

(6) 1 枚の納品書で発送する荷物の数は 10 個を超えないものとし，セル A16～G25 には，発送可能な組合せに分割した結果を表示する。また，セル B26～E26 及びセル G26 には，各列の行 16～25 の値の合計を表示する式を入力する。

〔マクロ：Package_count の説明〕

マクロ Package_count では，残重量が，発送可能な商品の総重量の上限を超える間は，その上限以下で最大の重量になるように荷物を作成し，残りを次の荷物に割り振る作業を繰り返すことによって組合せを求める。

(1) セル B4～D4 に入力されている数量を，セル B9～D9 に格納して残数量とする。また，荷物番号を数える変数 package_no に初期値として 1 を設定する。

(2) セル E9 の残重量が 0 になるまで，(3)～(5)の処理を繰り返す。

(3) 行 15 から数えて package_no の値だけ下の行を，該当する荷物番号の荷物を表示する行（以下，表示行という）とし，表示行の列 A に package_no，列 F にセル F9 の重量区分，列 G にセル G9 の発送料金を複写する。また，梱包する商品総重量の最大値を保存する変数 work_weight に，初期値として 0 を設定する。

(4) セル B10～D10 の数量を，それぞれ 0 から対応する商品の残数量まで順に変化させて，それぞれの数量の組合せで求めた作業重量の中で，値が一つの荷物として発送可能な商品の総重量の上限以下で最大となるものを表示行の列 E に，そのときの商品 X，Y 及び Z の数量を表示行の列 B～D に格納する。

(5) 商品の全ての組合せの比較が終了したとき，表示行の列 B～D に格納されている数量を，当該荷物番号の荷物に梱包する商品の数量として確定し，各商品の残数量から差し引き，package_no に 1 を加え，(2)に戻る。

〔マクロ：Package_count〕

```
○マクロ: Package_count
○数値型: package_no, i, j, k, m, work_weight
■ i: 0, i ≦ 2, 1
  ・相対(B9, 0, i) ← 相対(B4, 0, i)
■
・package_no ← 1
■ E9 > 0
  ・相対(A15, package_no, 0) ← package_no
  ・相対(F15, package_no, 0) ← F9
  ・相対(G15, package_no, 0) ← G9
  ・work_weight ← 0
  ■ i: 0, i ≦ B9, 1
    ・B10 ← i
    ■ j: 0, j ≦ C9, 1
      ・C10 ← j
      ■ k: 0, k ≦ D9, 1
        ・D10 ← k
        [ e ]
        ■ m: 0, m ≦ 3, 1
          ・相対(B15, package_no, m) ← 相対(B10, 0, m)
        ■
        ・work_weight ← E10
      ■
    ■
  ■
  ■ i: 0, i ≦ 2, 1
    ・[ f ]
  ■
  ・package_no ← package_no + 1
■
```

eに関する解答群

ア　論理積(work_weight ＜ E9, E10 ＞ 重量区分!E8)

イ　論理積(work_weight ＜ E10, E10 ≦ 重量区分!E8)

ウ　論理積(work_weight ＞ E9, E10 ＞ 重量区分!E8)

エ　論理積(work_weight ＞ E10, E10 ≦ 重量区分!E8)

fに関する解答群

ア　相対(B9, 0, i) ← 相対(B4, 0, i) − 相対(B10, 0, i)

イ　相対(B9, 0, i) ← 相対(B4, 0, i) − 相対(B15, package_no, i)

ウ　相対(B9, 0, i) ← 相対(B9, 0, i) − 相対(B10, 0, i)

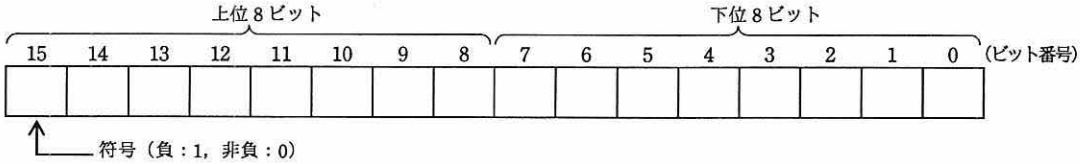
エ　相対(B9, 0, i) ← 相対(B9, 0, i) − 相対(B15, package_no, i)

■アセンブラ言語の仕様

1. システム COMETⅡの仕様

1.1 ハードウェアの仕様

(1) 1語は16ビットで，そのビット構成は，次のとおりである。



(2) 主記憶の容量は65536語で，そのアドレスは0～65535番地である。

(3) 数値は，16ビットの2進数で表現する。負数は，2の補数で表現する。

(4) 制御方式は逐次制御で，命令語は1語長又は2語長である。

(5) レジスタとして，GR（16 ビット），SP（16 ビット），PR（16 ビット），FR（3 ビット）の4種類がある。

GR（汎用レジスタ，General Register）は，GR0～GR7 の 8 個があり，算術，論理，比較，シフトなどの演算に用いる。このうち，GR1～GR7 のレジスタは，指標レジスタ（index register）としてアドレスの修飾にも用いる。

SP（スタックポインタ，Stack Pointer）は，スタックの最上段のアドレスを保持している。

PR（プログラムレジスタ，Program Register）は，次に実行すべき命令語の先頭アドレスを保持している。

FR（フラグレジスタ，Flag Register）は，OF（Overflow Flag），SF（Sign Flag），ZF（Zero Flag）と呼ぶ3個のビットからなり，演算命令などの実行によって次の値が設定される。これらの値は，条件付き分岐命令で参照される。

OF	算術演算命令の場合は，演算結果が－32768～32767 に収まらなくなったとき 1 になり，それ以外のとき0になる。論理演算命令の場合は，演算結果が0～65535 に収まらなくなったとき1になり，それ以外のとき0になる。
SF	演算結果の符号が負（ビット番号15が1）のとき1，それ以外のとき0になる。
ZF	演算結果が零（全部のビットが0）のとき1，それ以外のとき0になる。

(6) 論理加算又は論理減算は，被演算データを符号のない数値とみなして，加算又は減算する。

1.2 命令

命令の形式及びその機能を示す。ここで，一つの命令コードに対し 2 種類のオペランドがある場合，上段はレジスタ間の命令，下段はレジスタと主記憶間の命令を表す。

命令	書き方		命令の説明	FRの設定
	命令コード	オペランド		

(1) ロード，ストア，ロードアドレス命令

ロード LoaD	LD	r1,r2	r1 ← (r2)	○*1
		r,adr [,x]	r ← (実効アドレス)	
ストア STore	ST	r,adr [,x]	実効アドレス ← (r)	—
ロードアドレス Load ADdress	LAD	r,adr [,x]	r ← 実効アドレス	

(2) 算術，論理演算命令

算術加算 ADD Arithmetic	ADDA	r1,r2	r1 ← (r1) + (r2)	○
		r,adr [,x]	r ← (r) + (実効アドレス)	
論理加算 ADD Logical	ADDL	r1,r2	r1 ← (r1) $+_L$ (r2)	
		r,adr [,x]	r ← (r) $+_L$ (実効アドレス)	
算術減算 SUBtract Arithmetic	SUBA	r1,r2	r1 ← (r1) − (r2)	
		r,adr [,x]	r ← (r) − (実効アドレス)	
論理減算 SUBtract Logical	SUBL	r1,r2	r1 ← (r1) $-_L$ (r2)	
		r,adr [,x]	r ← (r) $-_L$ (実効アドレス)	
論理積 AND	AND	r1,r2	r1 ← (r1) AND (r2)	○*1
		r,adr [,x]	r ← (r) AND (実効アドレス)	
論理和 OR	OR	r1,r2	r1 ← (r1) OR (r2)	
		r,adr [,x]	r ← (r) OR (実効アドレス)	
排他的論理和 eXclusive OR	XOR	r1,r2	r1 ← (r1) XOR (r2)	
		r,adr [,x]	r ← (r) XOR (実効アドレス)	

(3) 比較演算命令

算術比較 ComPare Arithmetic	CPA	r1,r2	(r1) と (r2)，又は (r) と (実効アドレス) の算術比較又は論理比較を行い，比較結果によって，FR に次の値を設定する。	○*1
		r,adr [,x]		
論理比較 ComPare Logical	CPL	r1,r2		
		r,adr [,x]		

比較結果	FR の値 SF	FR の値 ZF
(r1) > (r2) (r) > (実効アドレス)	0	0
(r1) = (r2) (r) = (実効アドレス)	0	1
(r1) < (r2) (r) < (実効アドレス)	1	0

(4) シフト演算命令

算術左シフト Shift Left Arithmetic	SLA	r,adr [,x]	符号を除き (r) を実効アドレスで指定したビット数だけ左又は右にシフトする。 シフトの結果，空いたビット位置には，左シフトのときは 0，右シフトのときは符号と同じものが入る。	○*2
算術右シフト Shift Right Arithmetic	SRA	r,adr [,x]		
論理左シフト Shift Left Logical	SLL	r,adr [,x]	符号を含み (r) を実効アドレスで指定したビット数だけ左又は右にシフトする。 シフトの結果，空いたビット位置には 0 が入る。	
論理右シフト Shift Right Logical	SRL	r,adr [,x]		

(5) 分岐命令

正分岐 Jump on PLus	JPL	adr [,x]	FR の値によって，実効アドレスに分岐する。分岐しないときは，次の命令に進む。	—
負分岐 Jump on MInus	JMI	adr [,x]		
非零分岐 Jump on Non Zero	JNZ	adr [,x]		
零分岐 Jump on ZEro	JZE	adr [,x]		
オーバフロー分岐 Jump on OVerflow	JOV	adr [,x]		
無条件分岐 unconditional JUMP	JUMP	adr [,x]	無条件に実効アドレスに分岐する。	

命令	分岐するときの FR の値 OF	SF	ZF
JPL		0	0
JMI		1	
JNZ			0
JZE			1
JOV	1		

(6) スタック操作命令

プッシュ PUSH	PUSH　adr [,x]	SP ← (SP) $-_L$ 1, (SP) ← 実効アドレス	—
ポップ POP	POP　r	r ← ((SP)), SP ← (SP) $+_L$ 1	

(7) コール，リターン命令

コール CALL subroutine	CALL　adr [,x]	SP ← (SP) $-_L$ 1, (SP) ← (PR), PR ← 実効アドレス	—
リターン RETurn from subroutine	RET	PR ← ((SP)), SP ← (SP) $+_L$ 1	

(8) その他

スーパバイザコール SuperVisor Call	SVC　adr [,x]	実効アドレスを引数として割出しを行う。実行後の GR と FR は不定となる。	—
ノーオペレーション No OPeration	NOP	何もしない。	

注記　r，r1，r2　いずれも GR を示す。指定できる GR は GR0 ～ GR7

adr　アドレスを示す。指定できる値の範囲は 0 ～ 65535

x　指標レジスタとして用いる GR を示す。指定できる GR は GR1 ～ GR7

[　]　[　] 内の指定は省略できることを示す。

(　)　(　) 内のレジスタ又はアドレスに格納されている内容を示す。

実効アドレス　adr と x の内容との論理加算値又はその値が示す番地

←　演算結果を，左辺のレジスタ又はアドレスに格納することを示す。

$+_L$，$-_L$　論理加算，論理減算を示す。

FR の設定
- ○　：設定されることを示す。
- ○*1：設定されることを示す。ただし，OF には 0 が設定される。
- ○*2：設定されることを示す。ただし，OF にはレジスタから最後に送り出されたビットの値が設定される。
- —　：実行前の値が保持されることを示す。

1.3 文字の符号表

(1) JIS X 0201 ラテン文字・片仮名用 8 ビット符号で規定する文字の符号表を使用する。

(2) 右に符号表の一部を示す。1 文字は 8 ビットからなり，上位 4 ビットを列で，下位 4 ビットを行で示す。例えば，間隔，4，H，¥ のビット構成は，16 進表示で，それぞれ 20，34，48，5C である。16 進表示で，ビット構成が 21 ～ 7E（及び表では省略している A1 ～ DF）に対応する文字を図形文字という。図形文字は，表示（印刷）装置で，文字として表示（印字）できる。

(3) この表にない文字とそのビット構成が必要な場合は，問題中で与える。

行＼列	02	03	04	05	06	07
0	間隔	0	@	P	`	p
1	!	1	A	Q	a	q
2	”	2	B	R	b	r
3	#	3	C	S	c	s
4	$	4	D	T	d	t
5	%	5	E	U	e	u
6	&	6	F	V	f	v
7	’	7	G	W	g	w
8	(	8	H	X	h	x
9	)	9	I	Y	i	y
10	*	:	J	Z	j	z
11	+	;	K	[	k	{
12	,	<	L	¥	l	|
13	-	=	M	]	m	}
14	.	>	N	^	n	~
15	/	?	O	_	o	

2. アセンブラ言語 CASLⅡの仕様

2.1 言語の仕様

(1) CASLⅡは，COMETⅡのためのアセンブラ言語である。
(2) プログラムは，命令行及び注釈行からなる。
(3) 1命令は1命令行で記述し，次の行へ継続できない。
(4) 命令行及び注釈行は，次に示す記述の形式で，行の1文字目から記述する。

行の種類		記述の形式
命令行	オペランドあり	[ラベル] {空白} {命令コード} {空白} {オペランド} [{空白} [コメント]]
	オペランドなし	[ラベル] {空白} {命令コード} [{空白} [{;} [コメント]]]
注釈行		[空白] {;} [コメント]

注記 [ ] [ ] 内の指定が省略できることを示す。
{ } { } 内の指定が必須であることを示す。

ラベル　その命令の（先頭の語の）アドレスを他の命令やプログラムから参照するための名前である。長さは 1～8 文字で，先頭の文字は英大文字でなければならない。以降の文字は，英大文字又は数字のいずれでもよい。なお，予約語である GR0～GR7 は，使用できない。

空白　1文字以上の間隔文字の列である。

命令コード　命令ごとに記述の形式が定義されている。

オペランド　命令ごとに記述の形式が定義されている。

コメント　覚え書きなどの任意の情報であり，処理系で許す任意の文字を書くことができる。

2.2 命令の種類

命令は，4 種類のアセンブラ命令（START，END，DS，DC），4 種類のマクロ命令（IN，OUT，RPUSH，RPOP）及び機械語命令（COMETⅡの命令）からなる。その仕様を次に示す。

命令の種類	ラベル	命令コード	オペランド	機能
アセンブラ命令	ラベル	START	[実行開始番地]	プログラムの先頭を定義 プログラムの実行開始番地を定義 他のプログラムで参照する入口名を定義
		END		プログラムの終わりを明示
	[ラベル]	DS	語数	領域を確保
	[ラベル]	DC	定数 [,定数] …	定数を定義
マクロ命令	[ラベル]	IN	入力領域,入力文字長領域	入力装置から文字データを入力
	[ラベル]	OUT	出力領域,出力文字長領域	出力装置へ文字データを出力
	[ラベル]	RPUSH		GR の内容をスタックに格納
	[ラベル]	RPOP		スタックの内容を GR に格納
機械語命令	[ラベル]	（「1.2 命令」を参照）		

2.3 アセンブラ命令

アセンブラ命令は，アセンブラの制御などを行う。

(1)

START	[実行開始番地]

START 命令は，プログラムの先頭を定義する。

実行開始番地は，そのプログラム内で定義されたラベルで指定する。指定がある場合はその番地から，省略した場合は START 命令の次の命令から，実行を開始する。

また，この命令につけられたラベルは，他のプログラムから入口名として参照できる。

(2)

END	

END 命令は，プログラムの終わりを定義する。

(3)

DS	語数

DS 命令は，指定した語数の領域を確保する。

語数は，10 進定数（≧0）で指定する。語数を 0 とした場合，領域は確保しないが，ラベルは有効である。

(4)

DC	定数［,定数］…

DC 命令は，定数で指定したデータを（連続する）語に格納する。

定数には，10 進定数，16 進定数，文字定数，アドレス定数の 4 種類がある。

定数の種類	書き方	命令の説明
10 進定数	n	n で指定した 10 進数値を，1 語の 2 進数データとして格納する。ただし，n が−32768～32767 の範囲にないときは，その下位 16 ビットを格納する。
16 進定数	#h	h は 4 けたの 16 進数（16 進数字は 0～9，A～F）とする。h で指定した 16 進数値を 1 語の 2 進数データとして格納する（0000 ≦ h ≦ FFFF）。
文字定数	’文字列’	文字列の文字数（＞0）分の連続する領域を確保し，最初の文字は第 1 語の下位 8 ビットに，2 番目の文字は第 2 語の下位 8 ビットに，…と順次文字データとして格納する。各語の上位 8 ビットには 0 のビットが入る。 文字列には，間隔及び任意の図形文字を書くことができる。ただし，アポストロフィ（’）は 2 個続けて書く。
アドレス定数	ラベル	ラベルに対応するアドレスを 1 語の 2 進数データとして格納する。

2.4 マクロ命令

マクロ命令は，あらかじめ定義された命令群とオペランドの情報によって，目的の機能を果たす命令群を生成する（語数は不定）。

(1)

IN	入力領域,入力文字長領域

IN 命令は，あらかじめ割り当てた入力装置から，1 レコードの文字データを読み込む。

入力領域は，256 語長の作業域のラベルであり，この領域の先頭から，1 文字を 1 語に対応させて順次入力される。レコードの区切り符号（キーボード入力の復帰符号など）は，格納しない。格納の形式は，DC 命令の文字定数と同じである。入力データが 256 文字に満たない場合，入力領域の残りの部分は実行前のデータを保持する。入力データが 256 文字を超える場合，以降の文字は無視される。

入力文字長領域は，1 語長の領域のラベルであり，入力された文字の長さ（≧0）が 2 進数で格納される。ファイルの終わり（end of file）を検出した場合は，−1 が格納される。

IN 命令を実行すると，GR の内容は保存されるが，FR の内容は不定となる。

(2)

OUT	出力領域,出力文字長領域

OUT 命令は，あらかじめ割り当てた出力装置に，文字データを，1 レコードとして書き出す。

出力領域は，出力しようとするデータが 1 文字 1 語で格納されている領域のラベルである。格納の形式は，DC 命令の文字定数と同じであるが，上位 8 ビットは，OS が無視するので 0 でなくてもよい。

出力文字長領域は，1 語長の領域のラベルであり，出力しようとする文字の長さ（≧0）を 2 進数で格納しておく。

OUT 命令を実行すると，GR の内容は保存されるが，FR の内容は不定となる。

(3)

RPUSH	

RPUSH 命令は，GR の内容を，GR1，GR2，…，GR7 の順序でスタックに格納する。

(4)

RPOP	

RPOP 命令は，スタックの内容を順次取り出し，GR7，GR6，…，GR1 の順序で GR に格納する。

2.5 機械語命令

機械語命令のオペランドは，次の形式で記述する。

r，r1，r2	GR は，記号 GR0 ～ GR7 で指定する。
x	指標レジスタとして用いる GR は，記号 GR1 ～ GR7 で指定する。
adr	アドレスは，10 進定数，16 進定数，アドレス定数又はリテラルで指定する。 リテラルは，一つの 10 進定数，16 進定数又は文字定数の前に等号（=）を付けて記述する。CASLⅡは，等号の後の定数をオペランドとする DC 命令を生成し，そのアドレスを adr の値とする。

2.6 その他

(1) アセンブラによって生成される命令語や領域の相対位置は，アセンブラ言語での記述順序とする。ただし，リテラルから生成される DC 命令は，END 命令の直前にまとめて配置される。

(2) 生成された命令語，領域は，主記憶上で連続した領域を占める。

3. プログラム実行の手引

3.1 OS

プログラムの実行に関して，次の取決めがある。

(1) アセンブラは，未定義ラベル（オペランド欄に記述されたラベルのうち，そのプログラム内で定義されていないラベル）を，他のプログラムの入口名（START 命令のラベル）と解釈する。この場合，アセンブラはアドレスの決定を保留し，その決定を OS に任せる。OS は，実行に先立って他のプログラムの入口名との連係処理を行いアドレスを決定する（プログラムの連係）。

(2) プログラムは，OS によって起動される。プログラムがロードされる主記憶の領域は不定とするが，プログラム中のラベルに対応するアドレス値は，OS によって実アドレスに補正されるものとする。

(3) プログラムの起動時に，OS はプログラム用に十分な容量のスタック領域を確保し，その最後のアドレスに 1 を加算した値を SP に設定する。

(4) OS は，CALL 命令でプログラムに制御を渡す。プログラムを終了し OS に制御を戻すときは，RET 命令を使用する。

(5) IN 命令に対応する入力装置，OUT 命令に対応する出力装置の割当ては，プログラムの実行に先立って利用者が行う。

(6) OS は，入出力装置や媒体による入出力手続の違いを吸収し，システムでの標準の形式及び手続（異常処理を含む）で入出力を行う。したがって，IN，OUT 命令では，入出力装置の違いを意識する必要はない。

3.2 未定義事項

プログラムの実行等に関し，この仕様で定義しない事項は，処理系によるものとする。

表計算ソフトの機能・用語（基本情報技術者試験用）

表計算ソフトの機能，用語などは，原則として次による。

なお，ワークシートの保存，読出し，印刷，罫(けい)線作成やグラフ作成など，ここで示す以外の機能などを使用するときには，問題文中に示す。

1. ワークシート

(1) 列と行とで構成される升目の作業領域をワークシートという。ワークシートの大きさは 256 列，10,000 行とする。

(2) ワークシートの列と行のそれぞれの位置は，列番号と行番号で表す。列番号は，最左端列の列番号を A とし，A，B，…，Z，AA，AB，…，AZ，BA，BB，…，BZ，…，IU，IV と表す。行番号は，最上端行の行番号を 1 とし，1，2，…，10000 と表す。

(3) 複数のワークシートを利用することができる。このとき，各ワークシートには一意のワークシート名を付けて，他のワークシートと区別する。

2. セルとセル範囲

(1) ワークシートを構成する各升をセルという。その位置は列番号と行番号で表し，それをセル番地という。

［例］列 A 行 1 にあるセルのセル番地は，A1 と表す。

(2) ワークシート内のある長方形の領域に含まれる全てのセルの集まりを扱う場合，長方形の左上端と右下端のセル番地及び“：”を用いて，“左上端のセル番地：右下端のセル番地”と表す。これを，セル範囲という。

［例］左上端のセル番地が A1 で，右下端のセル番地が B3 のセル範囲は，A1:B3 と表す。

(3) 他のワークシートのセル番地又はセル範囲を指定する場合には，ワークシート名と“！”を用い，それぞれ“ワークシート名！セル番地”又は“ワークシート名！セル範囲”と表す。

［例］ワークシート“シート1”のセル B5 ～ G10 を，別のワークシートから指定する場合には，シート1!B5:G10 と表す。

3. 値と式

(1) セルは値をもち，その値はセル番地によって参照できる。値には，数値，文字列，論理値及び空値がある。

(2) 文字列は一重引用符“ ’ ”で囲って表す。

［例］文字列“A”，“BC”は，それぞれ 'A'，'BC' と表す。

(3) 論理値の真を true，偽を false と表す。

(4) 空値を null と表し，空値をもつセルを空白セルという。セルの初期状態は，空白セルとする。

(5) セルには，式を入力することができる。セルは，式を評価した結果の値をもつ。

(6) 式は，定数，セル番地，演算子，括弧及び関数から構成される。定数は，数値，文字列，論理値又は空値を表す表記とする。式中のセル番地は，その番地のセルの値を参照する。

(7) 式には，算術式，文字式及び論理式がある。評価の結果が数値となる式を算術式，文字列となる式を文字式，論理値となる式を論理式という。

(8) セルに式を入力すると，式は直ちに評価される。式が参照するセルの値が変化したときには，直ちに，適切に再評価される。

4. 演算子

(1) 単項演算子は，正符号“＋”及び負符号 “－”とする。

(2) 算術演算子は，加算“＋”，減算“－”，乗算“＊”，除算“／”及びべき乗“^”とする。

(3) 比較演算子は，より大きい“＞”，より小さい“＜”，以上“≧”，以下“≦”，等しい“＝”及び等しくない“≠”とする。

(4) 括弧は丸括弧“(”及び“)”を使う。

(5) 式中に複数の演算及び括弧があるときの計算の順序は，次表の優先順位に従う。

演算の種類	演算子	優先順位
括弧	( )	高
べき乗演算	^	↑
単項演算	＋，－	
乗除演算	＊，／	
加減演算	＋，－	↓
比較演算	＞，＜，≧，≦，＝，≠	低

5. セルの複写

(1) セルの値又は式を，他のセルに複写することができる。

(2) セルを複写する場合で，複写元のセル中にセル番地を含む式が入力されているとき，複写元と複写先のセル番地の差を維持するように，式中のセル番地を変化させるセルの参照方法を相対参照という。この場合，複写先のセルとの列番号の差及び行番号の差を，複写元のセルに入力された式中の各セル番地に加算した式が，複写先のセルに入る。

[例] セル A6 に式 A1 ＋ 5 が入力されているとき，このセルをセル B8 に複写すると，セル B8 には式 B3 ＋ 5 が入る。

(3) セルを複写する場合で，複写元のセル中にセル番地を含む式が入力されているとき，そのセル番地の列番号と行番号の両方又は片方を変化させないセルの参照方法を絶対参照という。絶対参照を適用する列番号と行番号の両方又は片方の直前には“$”を付ける。

[例] セル B1 に式 A1 ＋ $A2 ＋ A$5 が入力されているとき，このセルをセル C4 に複写す

ると，セル C4 には式 A1 ＋ $A5 ＋ B$5 が入る。

(4) セルを複写する場合で，複写元のセル中に，他のワークシートを参照する式が入力されているとき，その参照するワークシートのワークシート名は複写先でも変わらない。

［例］ワークシート“シート2”のセル A6 に式 シート1!A1 が入力されているとき，このセルをワークシート“シート3”のセル B8 に複写すると，セル B8 には式 シート1!B3 が入る。

6. 関数

式には次の表で定義する関数を利用することができる。

書式	解　　説
合計 (セル範囲[1])	セル範囲に含まれる数値の合計を返す。 ［例］合計 (A1:B5) は，セル A1 ～ B5 に含まれる数値の合計を返す。
平均 (セル範囲[1])	セル範囲に含まれる数値の平均を返す。
標本標準偏差 (セル範囲[1])	セル範囲に含まれる数値を標本として計算した標準偏差を返す。
母標準偏差 (セル範囲[1])	セル範囲に含まれる数値を母集団として計算した標準偏差を返す。
最大 (セル範囲[1])	セル範囲に含まれる数値の最大値を返す。
最小 (セル範囲[1])	セル範囲に含まれる数値の最小値を返す。
IF (論理式 , 式1, 式2)	論理式の値が true のとき式 1 の値を，false のとき式 2 の値を返す。 ［例］IF (B3 ＞ A4,' 北海道 ',C4) は，セル B3 の値がセル A4 の値より大きいとき文字列“北海道”を，それ以外のときセル C4 の値を返す。
個数 (セル範囲)	セル範囲に含まれるセルのうち，空白セルでないセルの個数を返す。
条件付個数 (セル範囲 , 検索条件の記述)	セル範囲に含まれるセルのうち，検索条件の記述で指定された条件を満たすセルの個数を返す。検索条件の記述は比較演算子と式の組で記述し，セル範囲に含まれる各セルと式の値を，指定した比較演算子によって評価する。 ［例1］条件付個数 (H5:L9, ＞ A1) は，セル H5 ～ L9 のセルのうち，セル A1 の値より大きな値をもつセルの個数を返す。 ［例2］条件付個数 (H5:L9, ＝ 'A4') は，セル H5 ～ L9 のセルのうち，文字列“A4”をもつセルの個数を返す。
整数部 (算術式)	算術式の値以下で最大の整数を返す。 ［例1］整数部 (3.9) は，3 を返す。 ［例2］整数部 (－3.9) は，－4 を返す。
剰余 (算術式1, 算術式2)	算術式1 の値を被除数，算術式2 の値を除数として除算を行ったときの剰余を返す。関数“剰余”と“整数部”は，剰余 (x,y) ＝ x － y ＊ 整数部 (x ／ y) という関係を満たす。 ［例1］剰余 (10,3) は，1 を返す。 ［例2］剰余 (－10,3) は，2 を返す。
平方根 (算術式)	算術式の値の非負の平方根を返す。算術式の値は，非負の数値でなければならない。
論理積 (論理式1, 論理式2, …)[2]	論理式1，論理式2，… の値が全て true のとき，true を返す。それ以外のとき false を返す。
論理和 (論理式1, 論理式2, …)[2]	論理式1，論理式2，… の値のうち，少なくとも一つが true のとき，true を返す。それ以外のとき false を返す。
否定 (論理式)	論理式の値が true のとき false を，false のとき true を返す。

切上げ（算術式，桁位置） 四捨五入（算術式，桁位置） 切捨て（算術式，桁位置）	算術式の値を指定した桁位置で，関数“切上げ”は切り上げた値を，関数“四捨五入”は四捨五入した値を，関数“切捨て”は切り捨てた値を返す。ここで，桁位置は小数第1位の桁を0とし，右方向を正として数えたときの位置とする。 ［例1］切上げ(−314.059,2) は，−314.06 を返す。 ［例2］切上げ(314.059, −2) は，400 を返す。 ［例3］切上げ(314.059,0) は，315 を返す。
結合(式1,式2,…)[2]	式1，式2，…のそれぞれの値を文字列として扱い，それらを引数の順につないでできる一つの文字列を返す。 ［例］結合('北海道','九州',123,456) は，文字列“北海道九州123456”を返す。
順位（算術式，セル範囲[1]，順序の指定）	セル範囲の中での算術式の値の順位を，順序の指定が0の場合は昇順で，1の場合は降順で数えて，その順位を返す。ここで，セル範囲の中に同じ値がある場合，それらを同順とし，次の順位は同順の個数だけ加算した順位とする。
乱数( )	0 以上 1 未満の一様乱数（実数値）を返す。
表引き(セル範囲，行の位置，列の位置)	セル範囲の左上端から行と列をそれぞれ1，2，…と数え，セル範囲に含まれる行の位置と列の位置で指定した場所にあるセルの値を返す。 ［例］表引き(A3:H11,2,5) は，セル E4 の値を返す。
垂直照合(式，セル範囲，列の位置，検索の指定)	セル範囲の左端列を上から下に走査し，検索の指定によって指定される条件を満たすセルが現れる最初の行を探す。その行に対して，セル範囲の左端列から列を1，2，…と数え，セル範囲に含まれる列の位置で指定した列にあるセルの値を返す。 ・検索の指定が0の場合の条件：式の値と一致する値を検索する。 ・検索の指定が1の場合の条件：式の値以下の最大値を検索する。このとき，左端列は上から順に昇順に整列されている必要がある。 ［例］垂直照合(15,A2:E10,5,0) は，セル範囲の左端列をセル A2，A3，…，A10 と探す。このとき，セル A6 で 15 を最初に見つけたとすると，左端列 A から数えて 5 列目の列 E 中で，セル A6 と同じ行にあるセル E6 の値を返す。
水平照合(式，セル範囲，行の位置，検索の指定)	セル範囲の上端行を左から右に走査し，検索の指定によって指定される条件を満たすセルが現れる最初の列を探す。その列に対して，セル範囲の上端行から行を1，2，…と数え，セル範囲に含まれる行の位置で指定した行にあるセルの値を返す。 ・検索の指定が0の場合の条件：式の値と一致する値を検索する。 ・検索の指定が1の場合の条件：式の値以下の最大値を検索する。このとき，上端行は左から順に昇順に整列されている必要がある。 ［例］水平照合(15,A2:G6,5,1) は，セル範囲の上端行をセル A2，B2，…，G2 と探す。このとき，15 以下の最大値をセル D2 で最初に見つけたとすると，上端行 2 から数えて 5 行目の行 6 中で，セル D2 と同じ列にあるセル D6 の値を返す。
照合検索(式，検索のセル範囲，抽出のセル範囲)	1 行又は 1 列を対象とする同じ大きさの検索のセル範囲と抽出のセル範囲に対して，検索のセル範囲を左端又は上端から走査し，式の値と一致する最初のセルを探す。見つかったセルの検索のセル範囲の中での位置と，抽出のセル範囲の中での位置が同じセルの値を返す。 ［例］照合検索(15,A1:A8,C6:C13) は，セル A1 ～ A8 をセル A1，A2，…と探す。このとき，セル A5 で 15 を最初に見つけたとすると，セル C6 ～ C13 の上端から数えて 5 番目のセル C10 の値を返す。

照合一致(式,セル範囲,検索の指定)	1行又は1列を対象とするセル範囲に対して，セル範囲の左端又は上端から走査し，検索の指定によって指定される条件を満たす最初のセルを探す。見つかったセルの位置を，セル範囲の左端又は上端から1，2，… と数えた値とし，その値を返す。 ・検索の指定が0の場合の条件：式の値と一致する値を検索する。 ・検索の指定が1の場合の条件：式の値以下の最大値を検索する。このとき，セル範囲は左端又は上端から順に昇順に整列されている必要がある。 ・検索の指定が −1 の場合の条件：式の値以上の最小値を検索する。このとき，セル範囲は左端又は上端から順に降順に整列されている必要がある。 ［例］照合一致(15,B2:B12,−1) は，セル B2 ～ B12 をセル B2，B3，…と探す。このとき，15 以上の最小値をセルB9 で最初に見つけたとすると，セル B2 から数えた値 8 を返す。
条件付合計(検索のセル範囲,検索条件の記述,合計のセル範囲[1)])	行数及び列数が共に同じ検索のセル範囲と合計のセル範囲に対して，検索と合計を行う。検索のセル範囲に含まれるセルのうち，検索条件の記述で指定される条件を満たすセルを全て探す。検索条件の記述を満たした各セルについての左上端からの位置と，合計のセル範囲中で同じ位置にある各セルの値を合計して返す。 検索条件の記述は比較演算子と式の組で記述し，検索のセル範囲に含まれる各セルと式の値を，指定した比較演算子によって評価する。 ［例1］条件付合計(A1:B8,＞E1,C2:D9) は，検索のセル範囲であるセル A1 ～ B8 のうち，セル E1 の値より大きな値をもつ全てのセルを探す。このとき，セル A2，B4，B7 が見つかったとすると，合計のセル範囲であるセル C2 ～ D9 の左上端からの位置が同じであるセル C3，D5，D8 の値を合計して返す。 ［例2］条件付合計(A1:B8,＝160,C2:D9) は，検索のセル範囲であるセル A1 ～ B8 のうち，160 と一致する値をもつ全てのセルを探す。このとき，セル A2，B4，B7 が見つかったとすると，合計のセル範囲であるセル C2 ～ D9 の左上端からの位置が同じであるセル C3，D5，D8 の値を合計して返す。

注[1)] 引数として渡したセル範囲の中で，数値以外の値は処理の対象としない。

[2)] 引数として渡すことができる式の個数は，1 以上である。

7. マクロ

(1) ワークシートとマクロ

ワークシートには複数のマクロを格納することができる。

マクロは一意のマクロ名を付けて宣言する。マクロの実行は，表計算ソフトのマクロの実行機能を使って行う。

［例］○マクロ: Pro

例は，マクロ Pro の宣言である。

(2) 変数とセル変数

変数の型には，数値型，文字列型及び論理型があり，変数は宣言することで使用できる。変数名にセル番地を使用することはできない。

［例］○数値型: row, col

例は，数値型の変数 row，col の宣言である。

セルを変数として使用でき，これをセル変数という。セル変数は，宣言せずに使用できる。

セル変数の表現方法には，絶対表現と相対表現とがある。

セル変数の絶対表現は，セル番地で表す。

セル変数の相対表現は，次の書式で表す。

書式	解　　説
相対(セル変数,行の位置,列の位置)	セル変数で指定したセルを基準のセルとする。そのセルの行番号と列番号の位置を 0 とし，下又は右方向を正として数え，行の位置と列の位置で指定した数と一致する場所にあるセルを表す変数である。

［例1］ 相対(B5, 2, 3) は，セル E7 を表す変数である。

［例2］ 相対(B5, −2, −1) は，セル A3 を表す変数である。

(3) 配列

数値型，文字列型又は論理型の配列は宣言することで使用できる。添字を“[”及び“]”で囲み，添字が複数ある場合はコンマで区切る。添字は 0 から始まる。

なお，数値型及び文字列型の変数及び配列の要素には，空値を格納することができる。

［例］ ○文字列型: table[100, 200]

例は， 100 × 200 個 の文字列型の要素をもつ 2 次元配列 table の宣言である。

(4) 宣言，注釈及び処理

宣言，注釈及び処理の記述は，“共通に使用される擬似言語の記述形式”の〔宣言，注釈及び処理〕に従う。

処理の記述中に式又は関数を使用する場合，その記述中に変数，セル変数又は配列の要素が使用できる。

［例］ ○数値型: row

```
row: 0, row < 5, 1
・相対(E1, row, 1) ← 垂直照合(相対(E1, row, 0), A1:B10, 2, 0) * 10
```

例は，セル E1，E2，…，E5 の各値に対して，セル A1 ～ A10 の中で同じ値をもつセルが現れる最初の行を探し，見つけた行の列 B のセルの値を10倍し，セル F1，F2，…，F5 の順に代入する。

〔 メ モ 用 紙 〕

〔 メ モ 用 紙 〕

〔 メ モ 用 紙 〕

〔 メ モ 用 紙 〕

6. 退室可能時間に途中で退室する場合には，手を挙げて監督員に合図し，答案用紙が回収されてから静かに退室してください。

退室可能時間	13:40 ～ 15:20

7. **問題に関する質問にはお答えできません。**文意どおり解釈してください。
8. 問題冊子の余白などは，適宜利用して構いません。ただし，問題冊子を切り離して利用することはできません。
9. アセンブラ言語の仕様及び表計算ソフトの機能・用語は，この冊子の末尾を参照してください。
10. 試験時間中，机上に置けるものは，次のものに限ります。
    なお，会場での貸出しは行っていません。
    受験票，黒鉛筆及びシャープペンシル（B又はHB），鉛筆削り，消しゴム，定規，時計（時計型ウェアラブル端末は除く。アラームなど時計以外の機能は使用不可），ハンカチ，ポケットティッシュ，目薬
    これら以外は机上に置けません。使用もできません。
11. 試験終了後，この問題冊子は持ち帰ることができます。
12. 答案用紙は，いかなる場合でも提出してください。回収時に提出しない場合は，採点されません。
13. 試験時間中にトイレへ行きたくなったり，気分が悪くなったりした場合は，手を挙げて監督員に合図してください。

試験問題に記載されている会社名又は製品名は，それぞれ各社又は各組織の商標又は登録商標です。
なお，試験問題では，™ 及び ® を明記していません。

©2017　独立行政法人情報処理推進機構